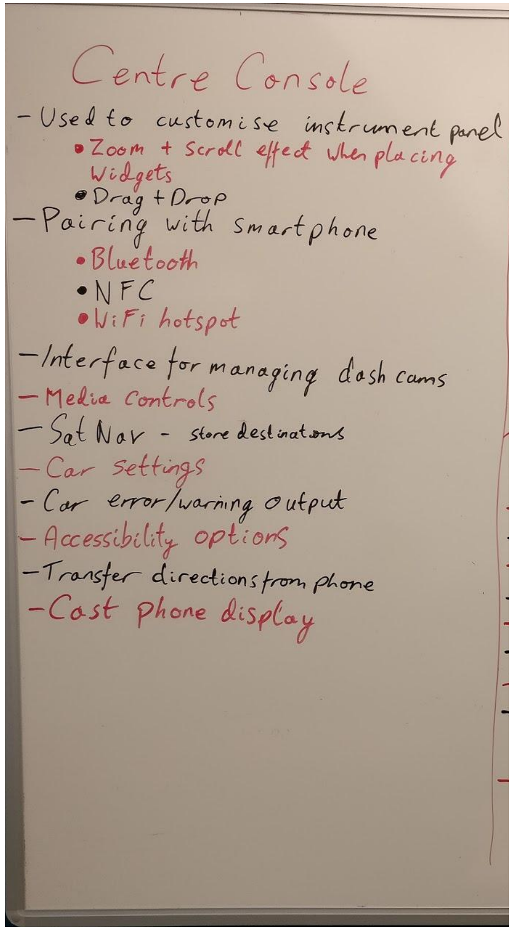
Dashboard Design

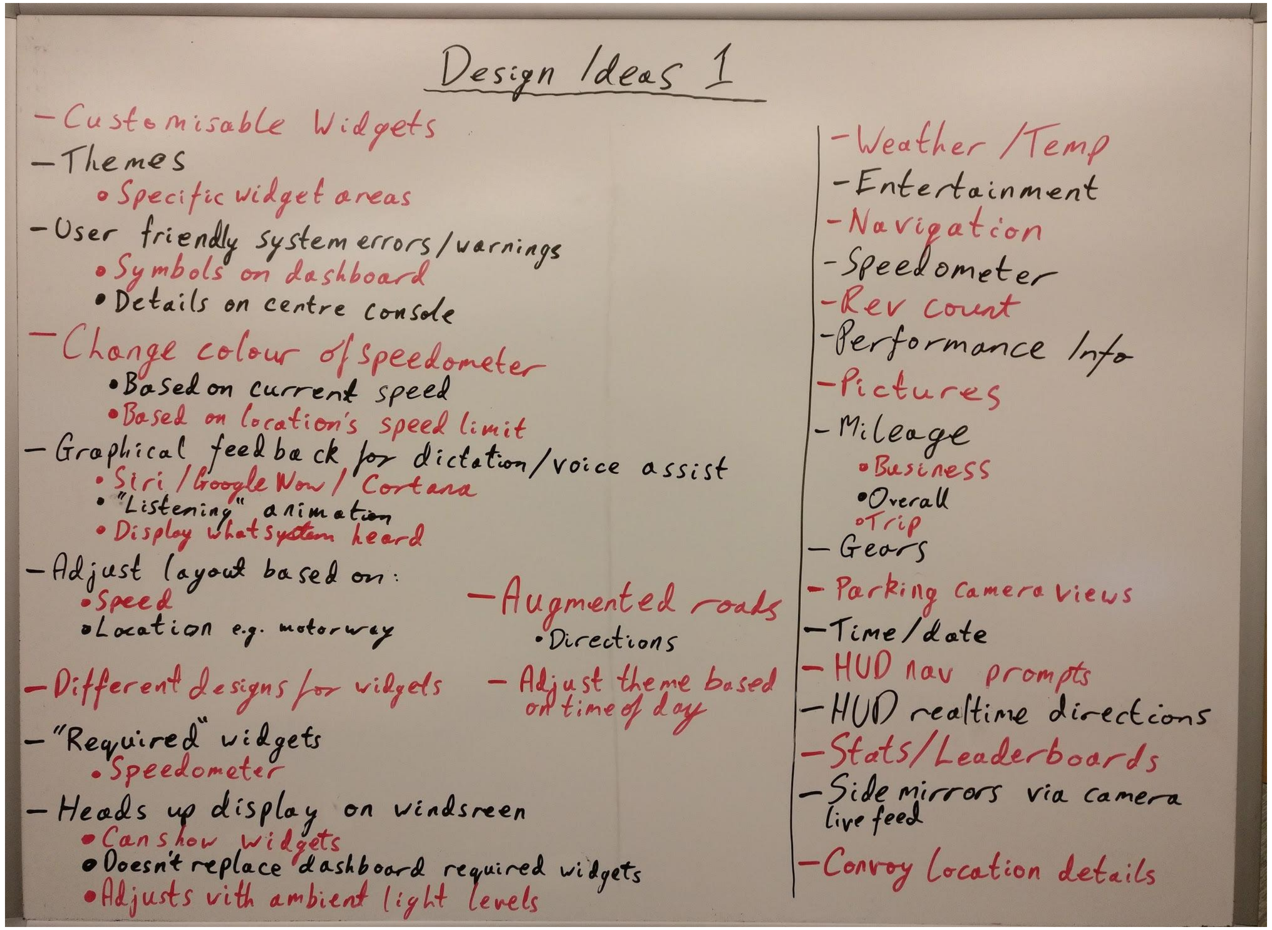
A-Team

Brainstorm

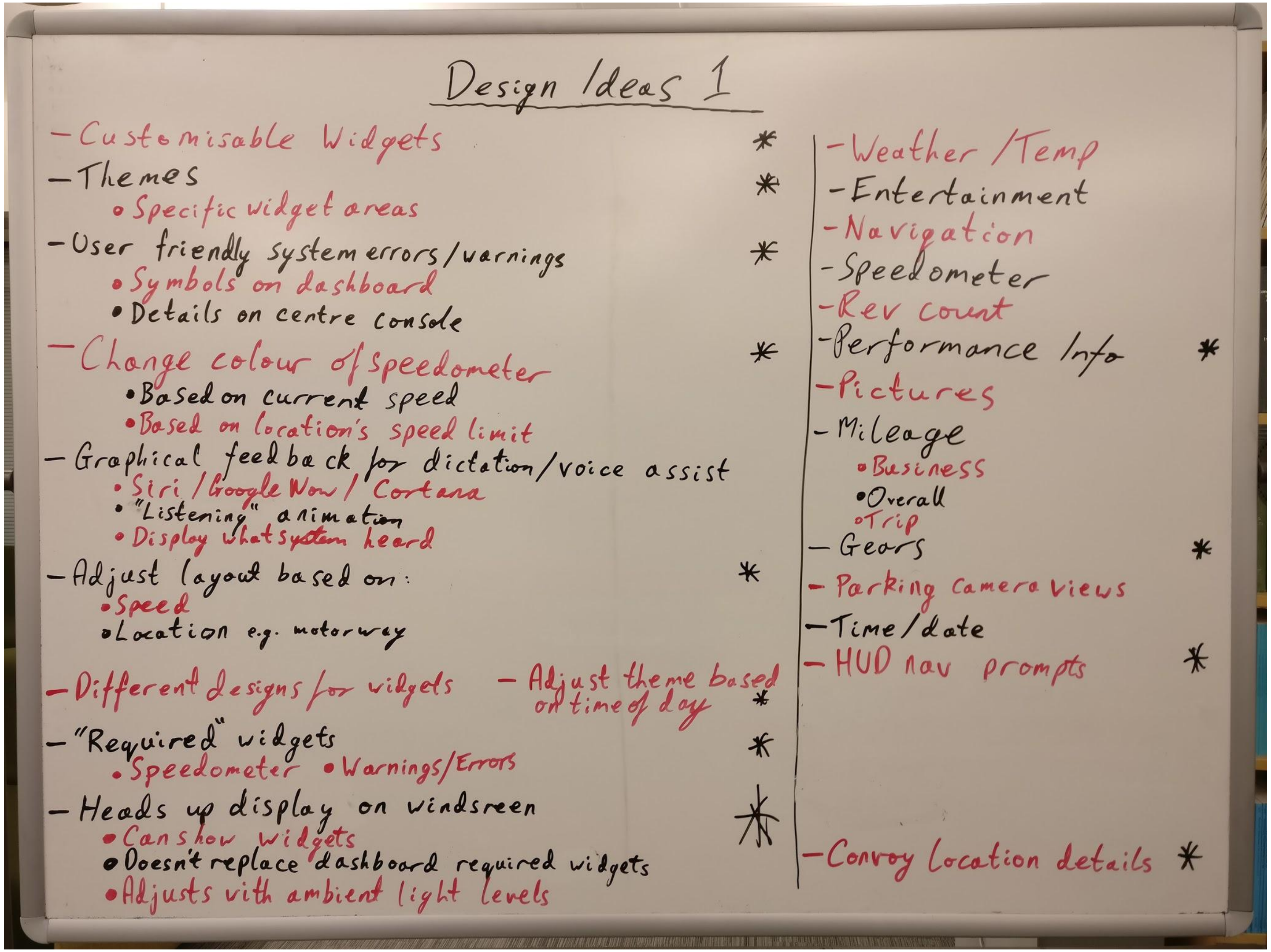
Centre console ideas



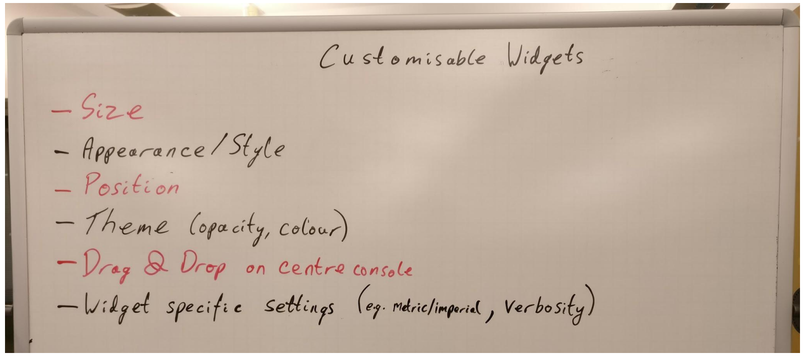
Instrument cluster ideas

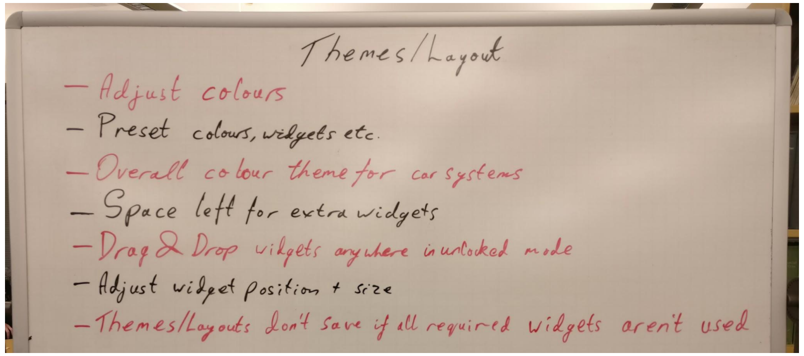


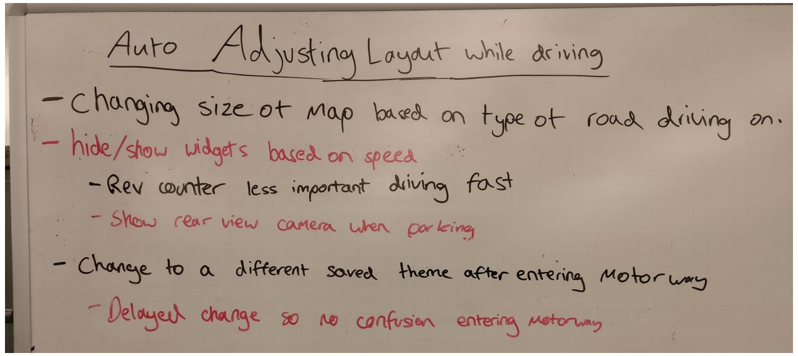
Brainstorm - Refined

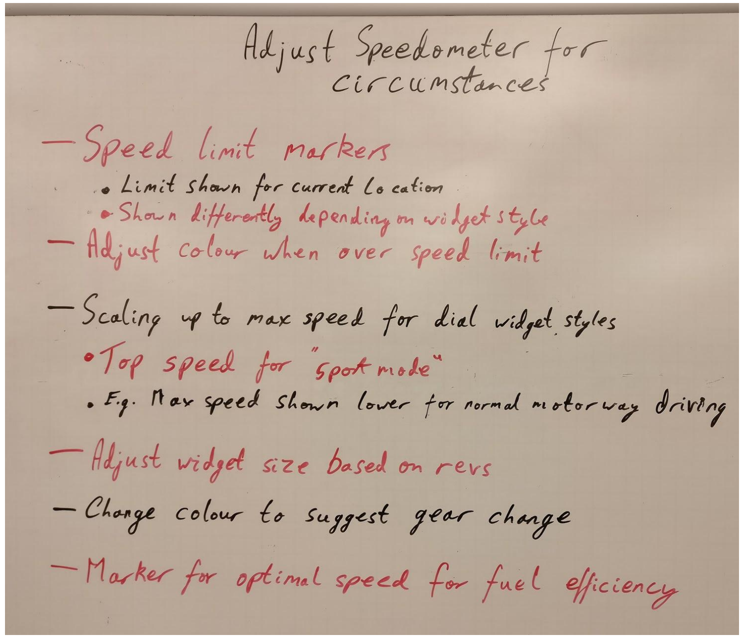


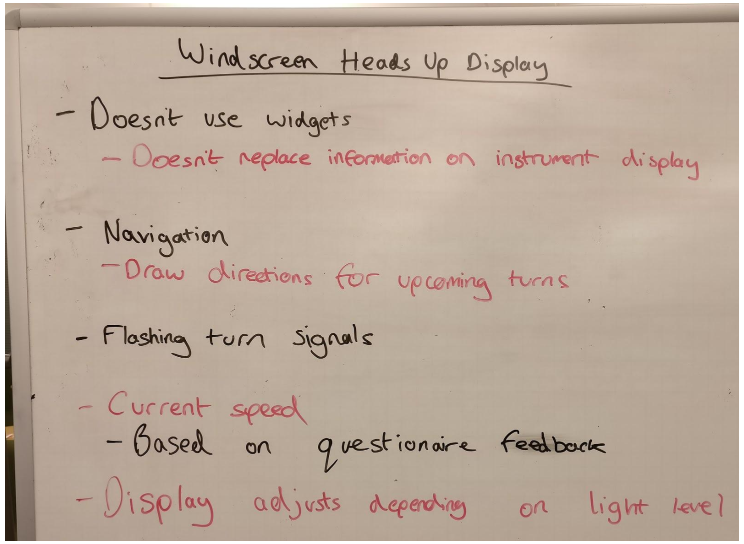
Brainstorm summary







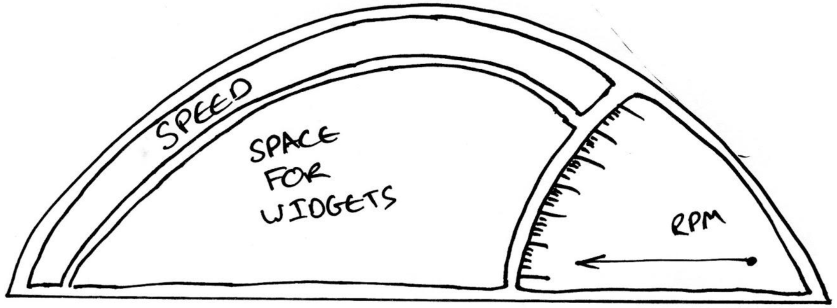




After getting all of our ideas down on the whiteboard, we ruled some of them out that we thought might not be suitable for our designs and starred the ideas that required elaboration.

Here you can see some of our elaboration slides, concluding our brainstorming.

Concept 1 - Dashboard Layouts

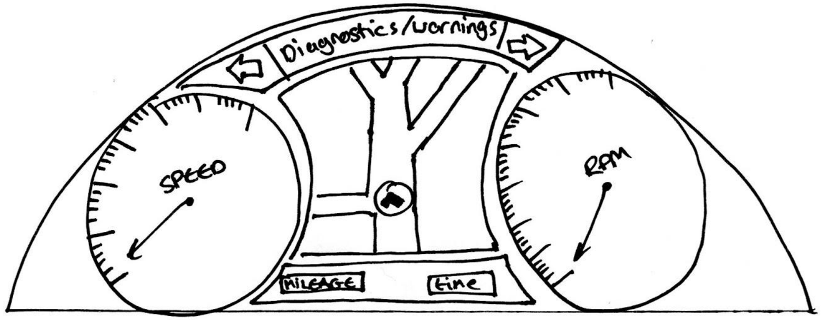


MORE ROOM FOR CUSTOMISATION INTERFACE

This interface is intended for users who really want to customise their dashboard. It features a much larger space to place widgets.

Different layouts can be selected to automatically switch to under certain circumstances; such as a motorway layout or a layout to use when the car has no internet connection

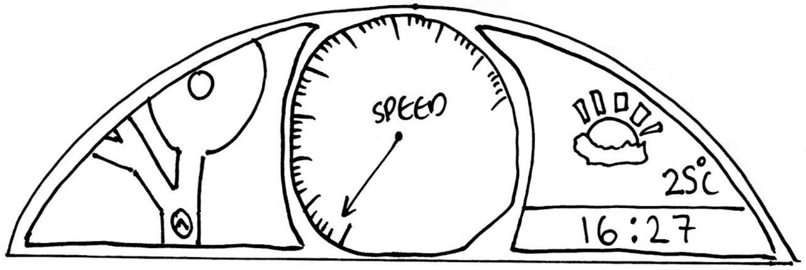
This interface was designed to look familiar to interfaces in current existing cars. The reason behind this is because the feedback we got from some of our questionnaires was that drivers didn’t want their dashboards changed. The centre of the screen is customisable with widgets.



STANDARD FAMILIAR INTERFACE

These are some examples of some preset layouts for the dashboard screen; the user can select one of these and have limited customisable space, or edit the layout from scratch.

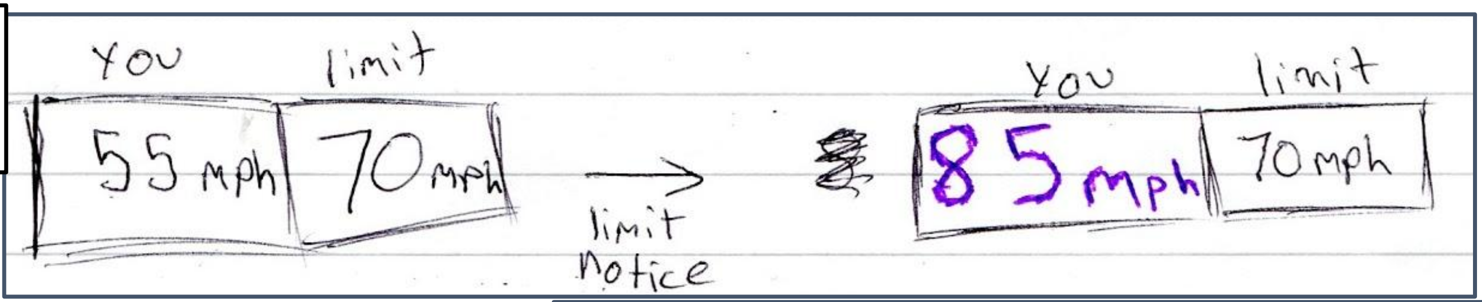
This interface is an alternate layout that splits the screen into two sections for customisation with widgets.



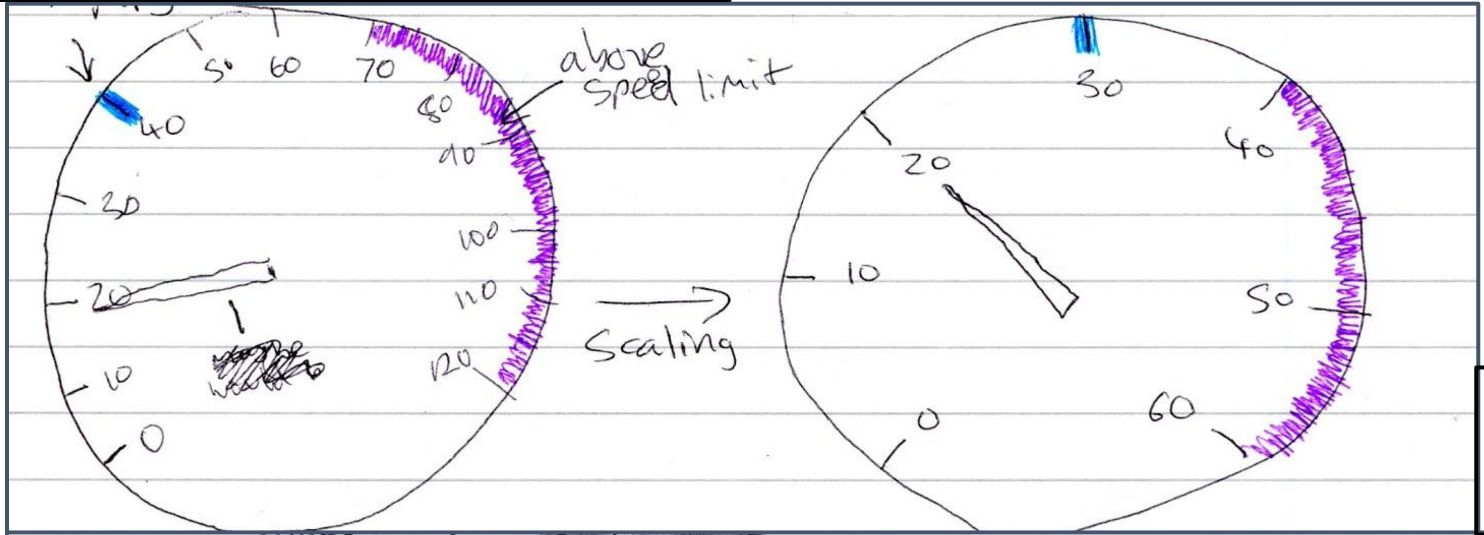
SPLIT SCREEN INTERFACE

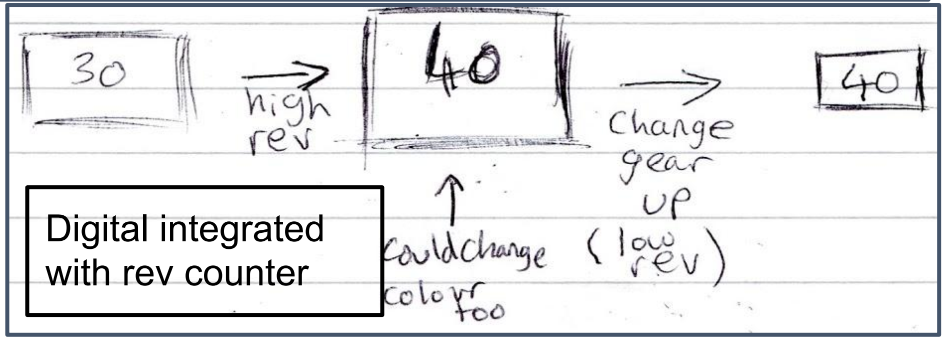
Concept 1 - Speedometer

Digital current speed and speed limit info



Standard dial with a marker for optimal speed for fuel efficiency and a marked zone for the speed limit

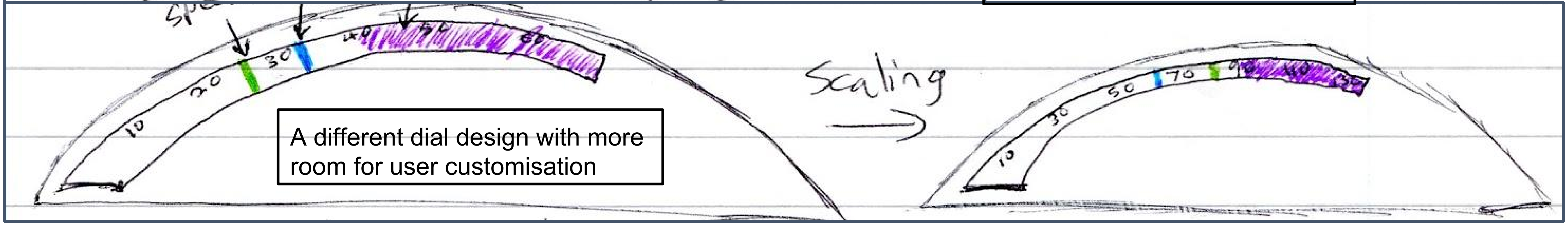




Digital integrated with rev counter

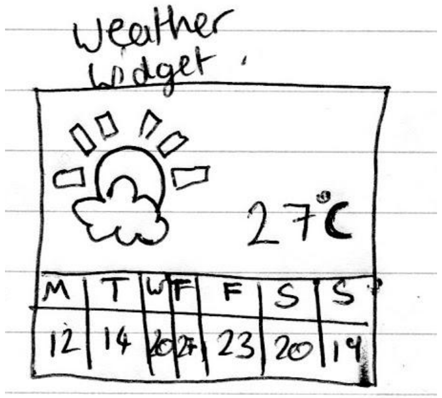
Speedometer gets larger as the rev count increases

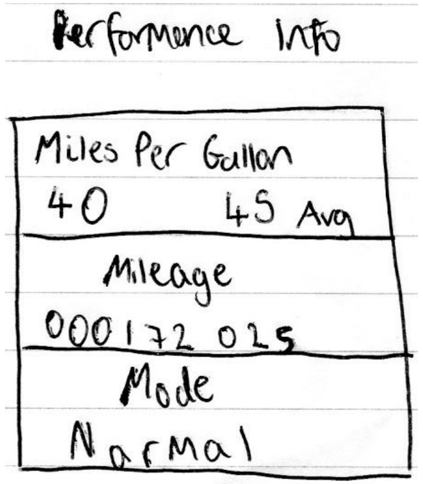
Dial scaling - the dial can change scales to suit current speed + speed limit

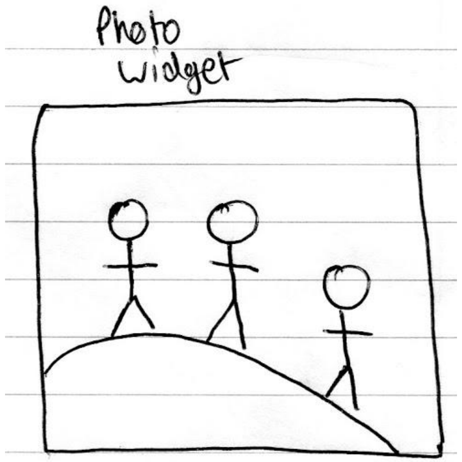


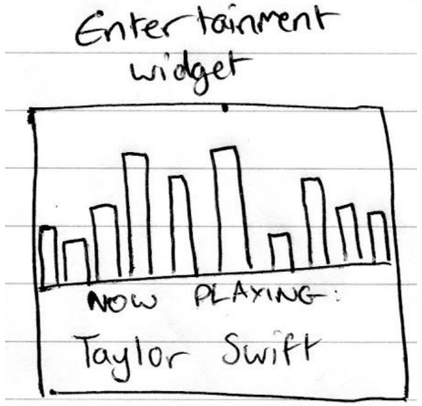
A different dial design with more room for user customisation

Concept 1 - Widgets

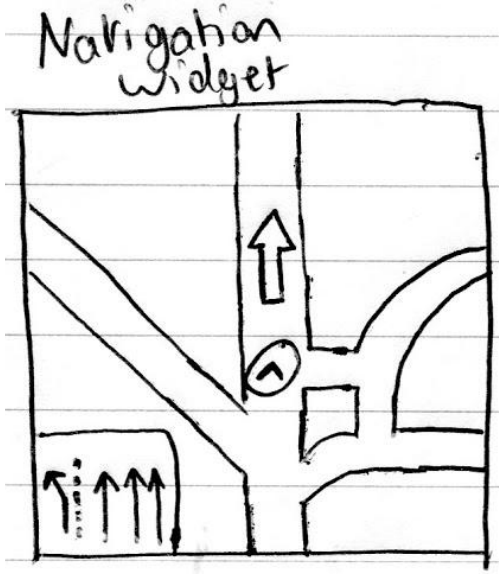


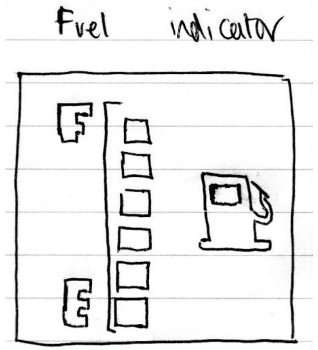


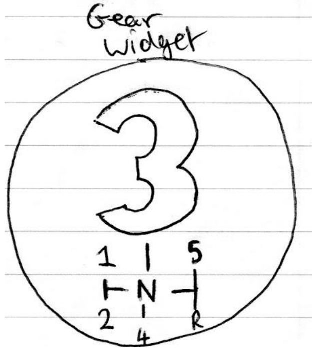


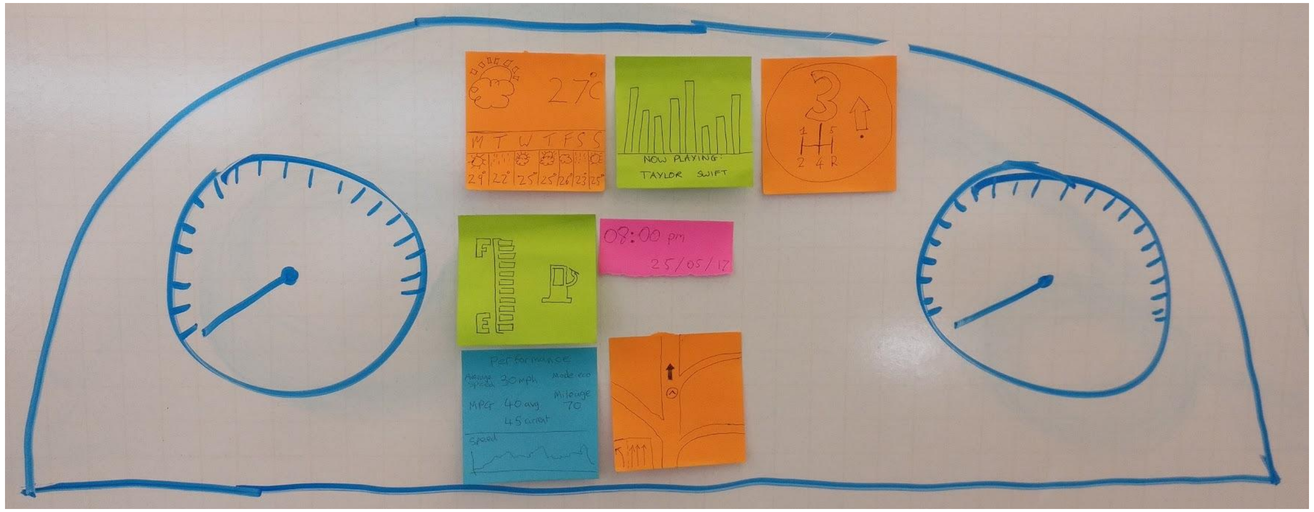


Here are some examples of widgets that could be available on the dashboard. Below is a layout of widgets including a fixed speedometer and RPM counter.



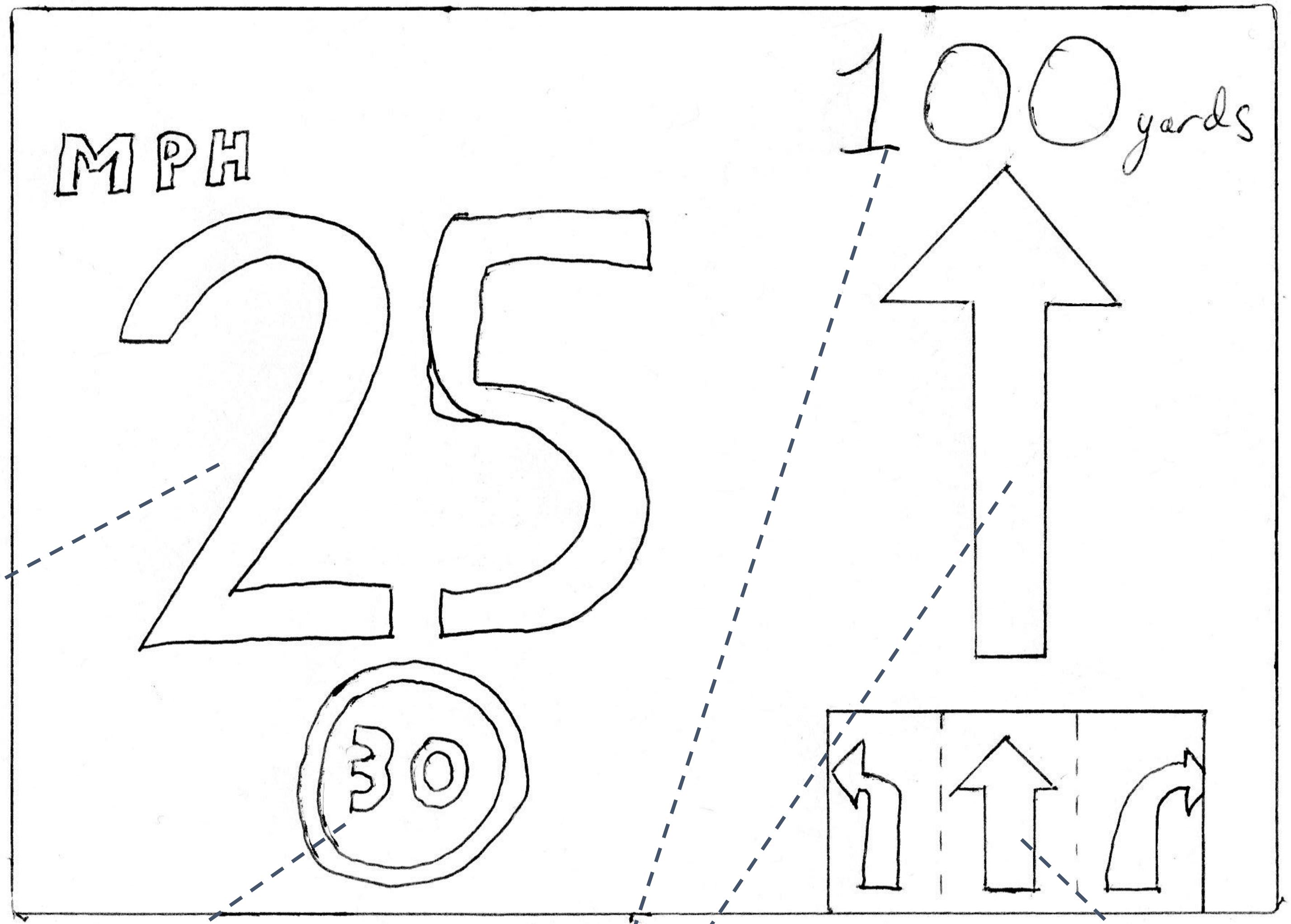






Concept 2

Our second concept consists of projecting the dashboard information onto the windscreen in the driver's view as a heads up display (HUD).



Large display of current speed.

Speed limit on current road.

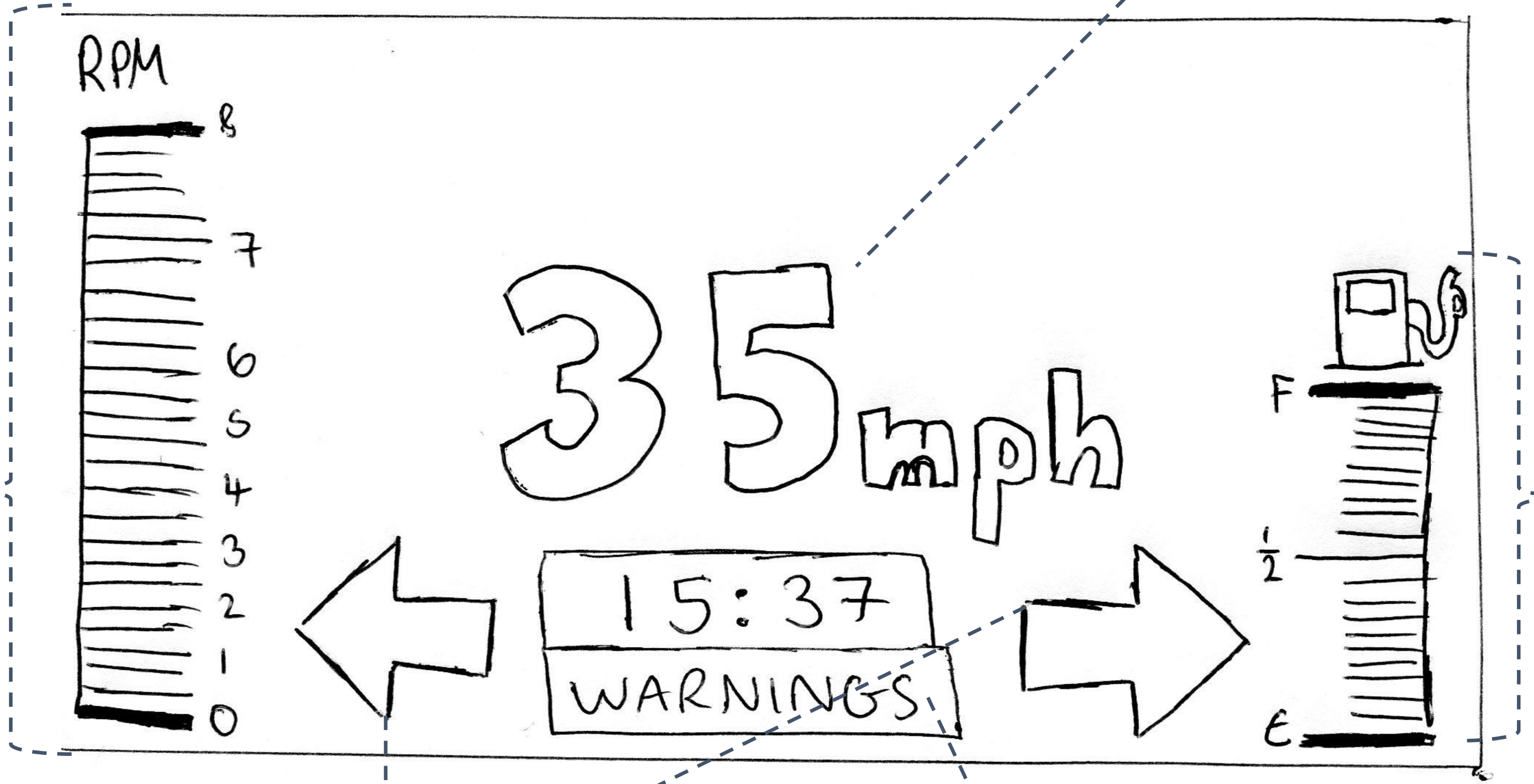
Direction and distance to turns for navigation route.

Highlight current lane to be in for navigation route.

Concept 2 - Design 2

Current speed. Units change based on preference settings.

Current revolutions per minute. As the revolutions increase a bar fills up and shoots back down after changing gears.



Fuel level indicator. Bar fills up based on quantity of fuel in tank.

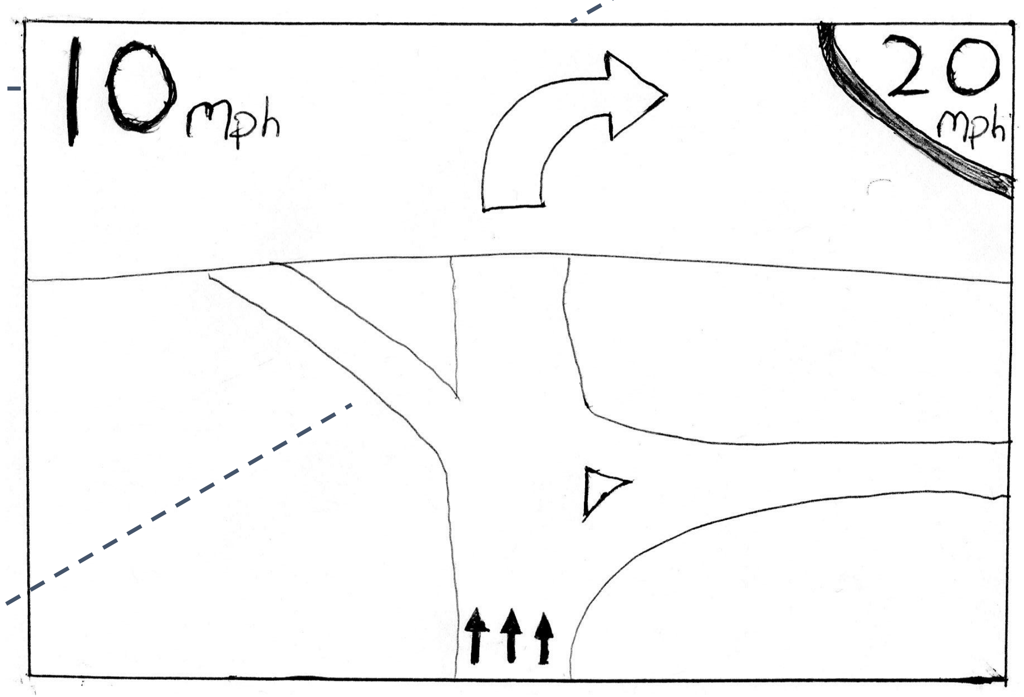
Indicator arrows, visual feedback if indicator is on.

Car warning lights. More information on each warning on the centre console.

Concept 2 - Design 3

The arrow at the top shows the upcoming SatNav instruction.

The top left number is the car's current speed.



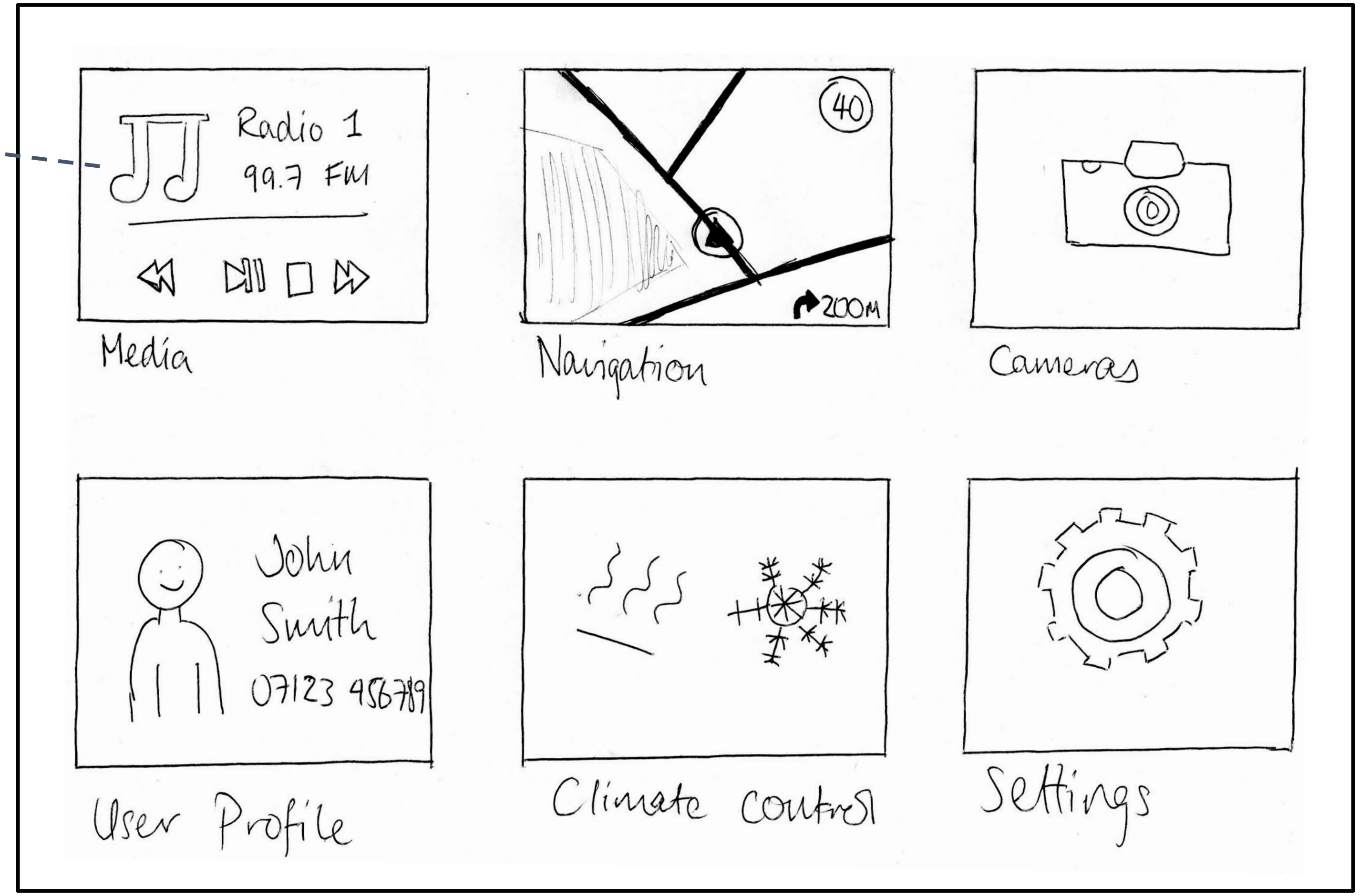
The number in the top right shows the current road's speed limit. This number will be surrounded by a red line so that it is similar to the UK speed limit signs.

This design for the HUD differs from our previous ones most in that it has a display of the road ahead for it's directions.

This allows for detailed directions that can be understood by the user without having to look away from the road.

Concepts 1 & 2 - centre Console

Large buttons with contextual information



Both concepts would also have a centre console included. Since our first concept heavily relies on using the centre console for customising via drag and dropping widgets, we have included a lo-fi prototype of what the centre console could look like.

Final version

Weather information widget. Displays temperature information of the whole week and forecasts for the current day

Navigation widget

Traditional dial displays - Optional apart from speedometer

Temperature dial. The colouring of the dial is a visual indicator of the state of the coolant as:
Blue - Cold
Green - Optimal operation temperature
Red - Overheating

Speed dial. Green shows optimal speed for fuel efficiency and red is outlining speeds over the speed limit which is 70 in this case

Final design

Large modern speedometer

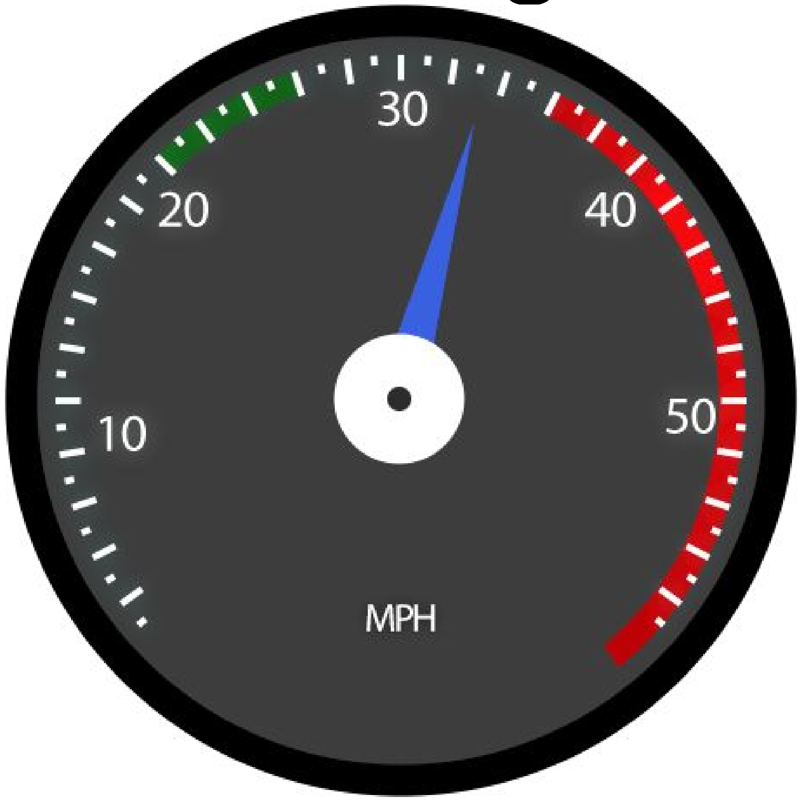
Entertainment widget. It shows the current song playing and a visualisation synchronised to the music playing

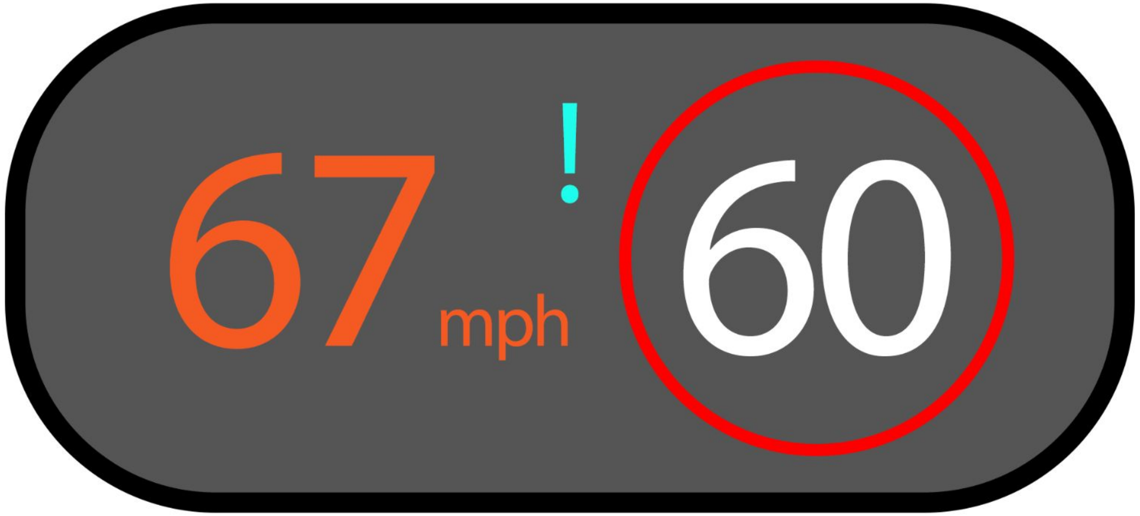
Modern displays of temperature and fuel level, color coded similarly to the ones with analogue styling

Gear widget.

Final design

These designs are animated to show how they change for given situations.These do not reliably show properly on all computers, so to see the animations, go to http://imgur.com/a/dbio5





The driver can choose a digital display, with some further optional features shown here; a speed limit notifier including colour and (!) warnings while exceeding, as well as an integrated rev counter, changing the size of the speedometer widget based on current rev count.

We had the idea that an analogue dial-style display could potentially scale itself to suit different conditions, for example as the car speeds up or as the speed limit changes.

This, however, brought up the issue of what to do with the arrow - it might be concerning to the driver to suddenly notice the arrow making a drastic move. As a result we tested the idea of phasing the arrow to a new location. This idea may still be too distracting for the driver, which we need to test.

Final design - Centre console

Large buttons which are easy to click on, even on a 10” display

Tiles live-update with relevant info like GPS location or camera images

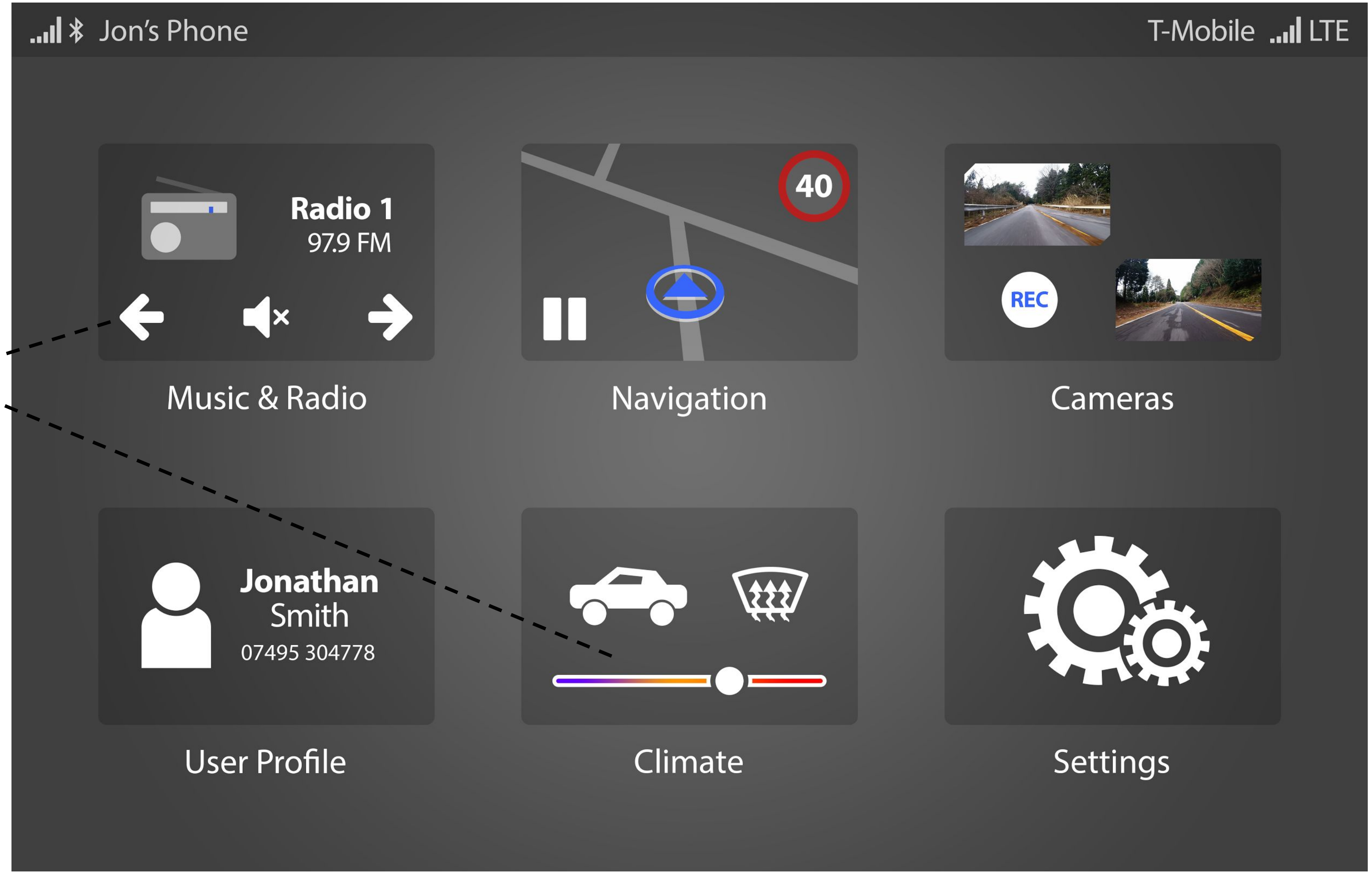
Phone connected via bluetooth for syncing profiles & making calls/web

Each button can have several ‘quick-action’ buttons (or sliders etc) that can be activated from the home screen

Icons that are missing from this design are:

- Convoy
- Phone/SMS

and likely other useful features.



The settings icon will reveal a submenu with, but not limited to, the following options:

```
- Accessibility
- Customisation
  * Instrument cluster
  * Colours
  * Themes
  * Icons
  * Home screen
- Car settings
  * Handling mode
    (sporty/city/etc)
- Connectivity
  * Bluetooth
  * WiFi
  * NFC
  * Cellular connection
  * Phone tether
```

Comments

We found this project rather tricky at times, with requirements being unclear, likely on purpose. We were also unsure on what tasks lay ahead for future weeks and so we were unable to plan for future weeks.

Since the requirements were not clear, we decided to start our project by researching current dashboard systems available and then drawing from that a questionnaire that would be able to determine what consumers want on their dashboard and what can be removed.

The questionnaires made our requirements clearer and so we started collecting ideas on a whiteboard for what features we can plan and design.

Since some of our initial designs were well produced, we often found it hard to produce visibly higher quality designs later in the project.

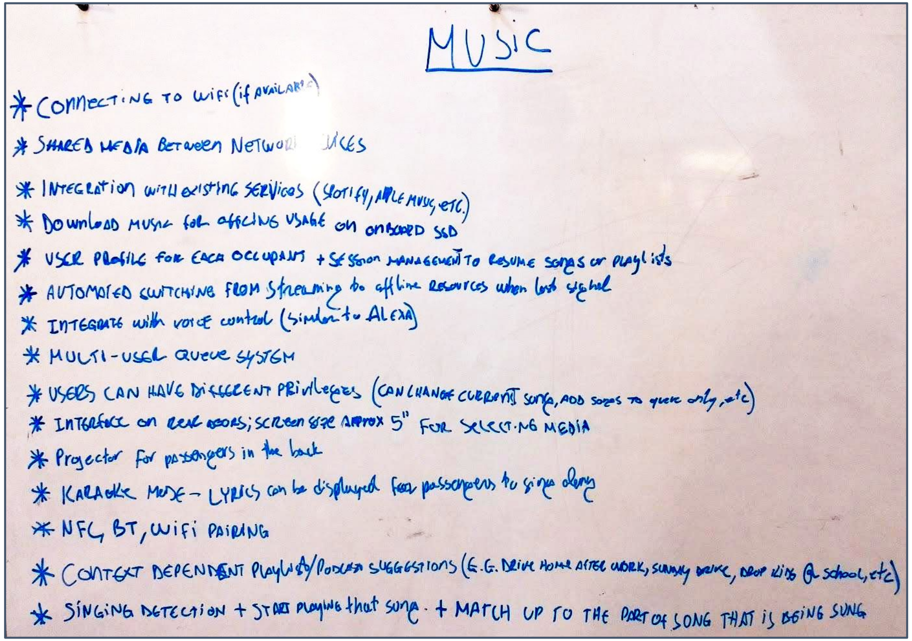
We're happy with our final designs but we believe that some elements could still be perfected. If we had more time available then we would likely have not created the final design just yet but taken some of our initial designs to be reviewed by consumers.

Following the diamond model for our project made decisions and planning much more approachable tasks and allowed us to work effectively as a team.

Music System Design

A Team

Brainstorm



Our initial brainstorming was done on a whiteboard, writing down any ideas that we were able to come up with for car music system features.

We were able to come up with new, innovative ideas for the system, such as the car being able to detect what song is being sung by a person and then play the most popular version of that song; context dependent playlist suggestions; and having a multi-user queue system .

The brainstorm session went well and allowed us to create a base of ideas to build upon for further research, questionnaires and personas.

Brainstorm summary

- Integrate with voice control (similar to Alexa) (**Chad**)
  - Make use of existing NLP libraries, very extensive work to construct from scratch
  - Give the user an intuitively accessible, easy option to set the car/entertainment systems' name (Does not seem to exist in industry, yay!)
  - Potentially needs some training with each users voice

- Multi-user queue system (**Geoff**)
  - Queue stored locally
  - Each time a song is added to the queue is immediately buffered if any bandwidth is available or added to a download queue of songs to be buffered if one exists
  - Songs can be added through any of the screens in the car, multifunctional steering wheel, voice control and hand gestures if available.

- Users can have different privileges (can change current song, add songs to queue only) (**Agatha**)
  - Low privilege users can request songs which can be accepted or denied
  - Medium privilege users can add songs to queue but not change current song
  - High privilege users (normally the driver) can do all of the above
  - Set lower privileges by seat? (Rear seats get lower priority, front passenger gets medium and Driver gets high as a default or something similar)
- Interface on rear doors (**Geoff's kids, Tony's kids and Paul's kids**)
  - 5" screen
  - Used for selecting media
  - Not always on, to prevent accidentally changing media (Could be locked/unlocked like a phone?)
  - Allows for limited, simplistic functionality available for media in the centre console (Search & choose songs, skip, add to queue, pause etc.)
- Karaoke mode - lyrics can be displayed for passengers to sing along (**Agatha, Geoff's kids**)

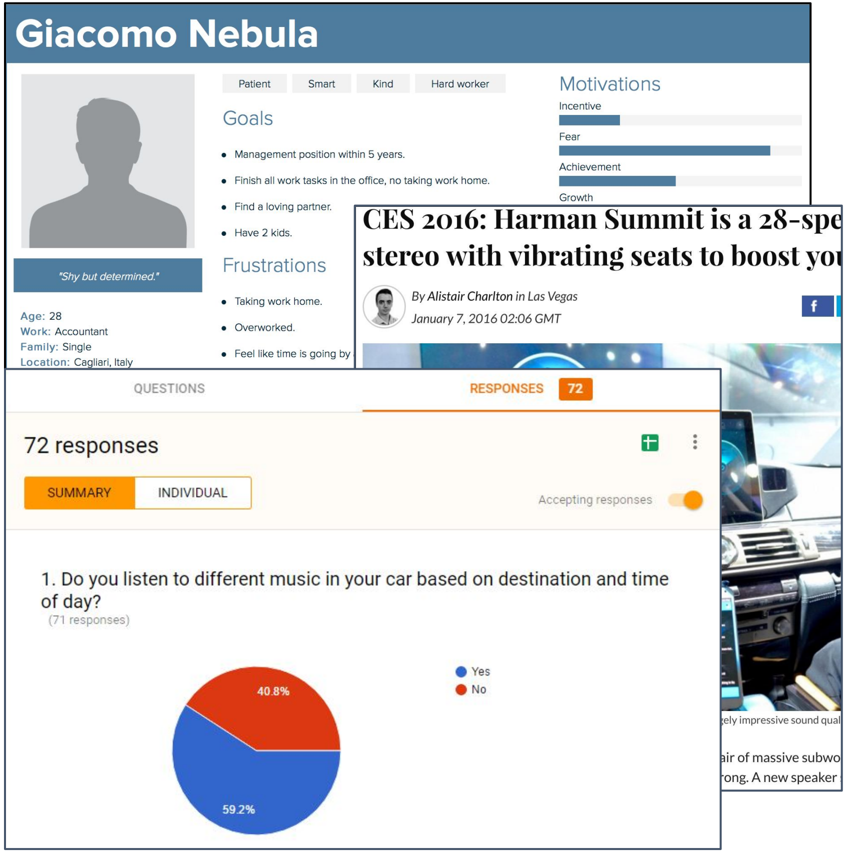
Once we had collected all of our ideas for a car music system, we ruled out ideas that we thought might be infeasible - such as the singing detection, karaoke mode, and back-seat projectors - and then grouped similar ideas.

We expanded on each idea to gain some more detailed insights as to what might be required to implement them. We also tagged each feature with names of personas we had created that they might be especially suited to, this allowed us to design the system taking each person into account.

Since we were going into details about the system's features, we tried to be specific about what kind of technical requirements would exist for each, such as security.

With more detailed ideas, we started to get a feel for how our concepts might come together and how our system would be an improvement upon existing systems.

Research



Once we ran out of ideas in the brainstorming phase, we created 12 personas that were all different. Personas were created in order to inspire more ideas of features for each user of the car. We covered many details for each persona such as hobbies, family and day to day activities. The personas allowed us to imagine scenarios to inspire improvements to current ideas and generate new ones.

Once we felt happy with the features that we had collected, we created an online questionnaire to validate our ideas, seek improvements and discover new ideas. We received over 70 responses and were able to understand popular opinions that were different to what we had expected.

We also researched current existing technologies demonstrated in the latest car shows to ensure that we were being as innovative and up-to-date as possible with what we planned to design.

Link to the questionnaire results: http://bit.ly/2lgzvGu

Shared Core Concepts

Both of our concepts share these features:

- SSD storage embedded inside of the car to store media locally
- Shared media between home network devices
  - Automatic home network sync over WiFi
  - Requires strong security for WiFi networks e.g. WPA2 with AES
  - Devices must be “trusted” before they can communicate with the media system
- Streamed media is cached for later playback
  - Remove less commonly played songs when cache is full
- Automated switching from streaming to offline resources when internet connection lost
- Context dependant playlist/podcast suggestions
  - Saving playlists to user preference profile alongside context, for automatic resuming of media
    - For example, automatic detection that a person listens to a specific playlist when driving home from work
- User profile system synchronised with the cloud to store database of preferences and details of previous sessions
  - System to identify a user - a way to log in (through one of or a combination of the following)
    - Phone connection (with your profile ID stored on your phone)
    - Fingerprint
    - Weight on seat
    - Driver could be identified through an ID on the key

Concept 1 - Key features

Our first concept is based on the idea of each seat having its own individual sound output so that each person can listen to their own music.

- Each car seat will have its own speakers embedded into the headrest where sound is only audible when your head is present in the headrest region.
- Passengers in the front can control their music via the centre console. Back passengers can do this via 5” interfaces located in the armrest of each door or their mobile phones.
- Additional features include a voice interaction system for the driver similar to Siri/Cortana/Alexa, to allow hands-free editing of their own playlist

Concept 1

When we first came up with the idea for this concept we were discouraged as we thought it was a bit too far fetched. How could you not be able to hear other passengers' personal audio? It was like something you would see in a futuristic movie.

We thought that giving each passenger personal headphones might be a viable option, but the driver would not be allowed to wear headphones for safety reasons and also, the issue of this being dangerous due to placement of wires arose.

However, during our research we found that BMW had developed something called the "Personal BMW Sound Curtain" which allows the driver and passengers listen to different music at the same time. BMW implemented this concept into a working prototype which was showcased at the CES 2017 event *[1].

We were glad that this is already an emerging technology that we could look to have built into our entertainment system. Car passengers do not always agree on the music that is playing in the car and so this could be a big attraction for consumers that want to listen to their own music and not others'.

[1] http://www.bmwblog.com/2017/01/04/bmw-unveiled-its-bmw-i-inside-future-sculpture-at-ces-2017/

Concept 1

Here are the lo-fi designs that we created for our first concept.

The speakers can be seen in the headrest of the seat, in a racing seat style that wraps slightly around the head of the passenger.

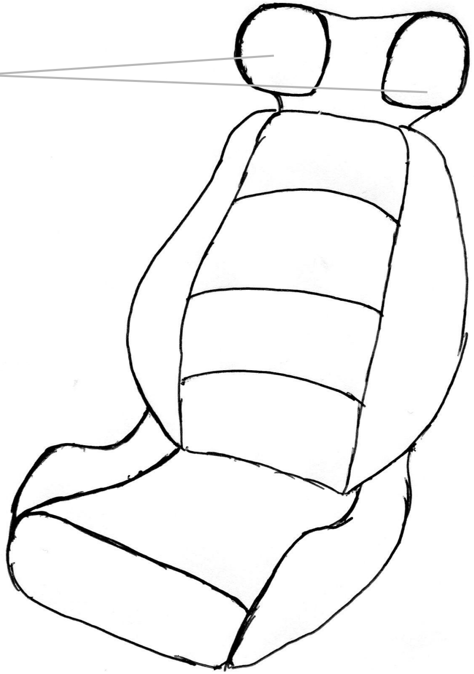
The speakers are designed and directed so that only the person sitting in their seat can hear their own music.

The speakers offer noise cancellation to reduce the ambient noise of the car while driving, this is cut out when important sounds are heard, i.e. car horns / sirens

Since only the driver requires the SatNav instructions, the system defaults to only outputting audio based instructions to the driver's speakers.

Surround sound speakers

Top-down view of seat with passenger



Front view of empty seat

Disadvantages of concept:
- Likely only to work when you're sat in a specific position
- May not be as loud as you want it
- May obstruct view from driver
- Reduces social aspect of group driving

Advantages of concept:
+ Each person can listen to what they want
+ Passengers aren't distracted by satnav commands

Concept 2 - Key features

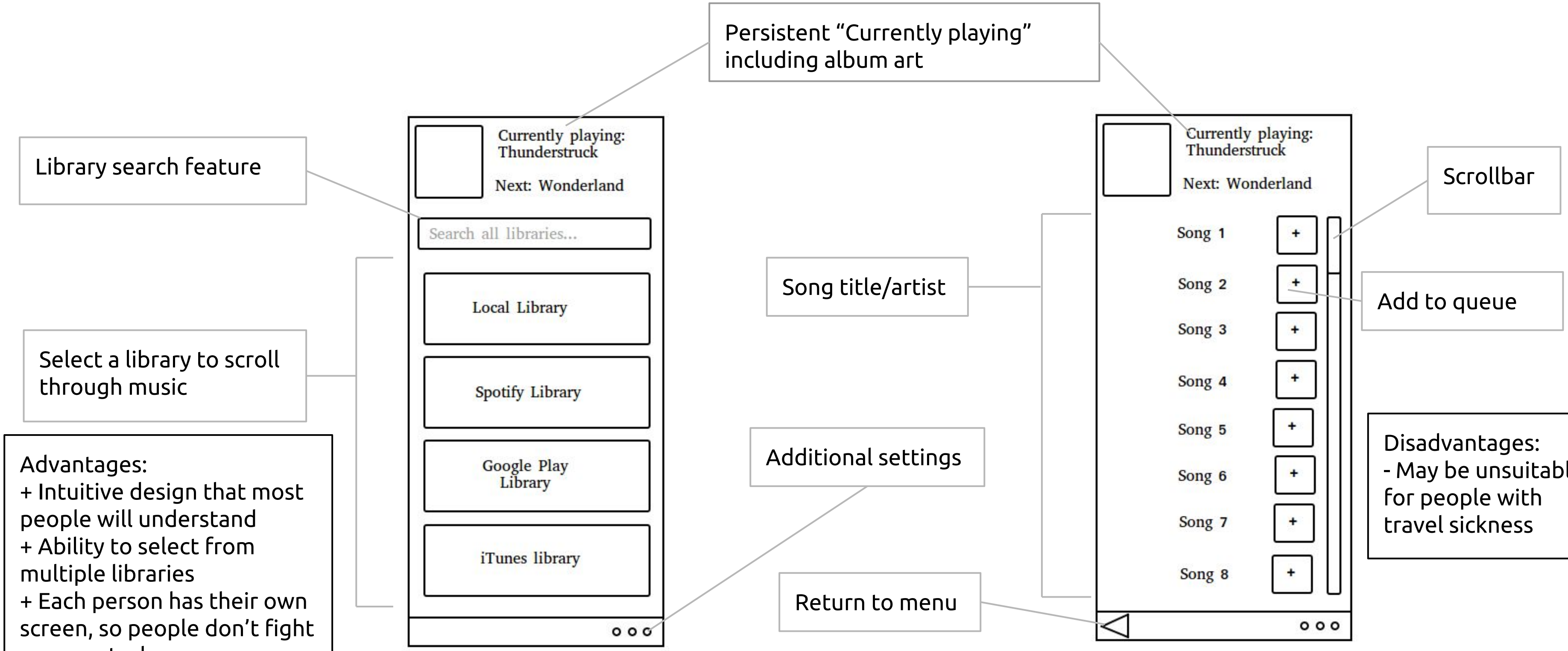
Our second concept is based on the idea of having a shared playlist or queue where passengers can select their own music and add it.

- Each passenger has privileges set by the driver. Privileges allow for control of which seats can add songs without confirmation, who can skip songs etc.
- The queue will be stored locally and restore for each journey unless cleared.
- Queued songs are buffered ahead of play time if the music is not already stored locally.
- The queue can contain music from many different services (Spotify, Google Play, local) using external APIs *[2]

[2] https://github.com/watsonbox/exportify

Concept 2

Here is a lo-fi design prototype for the music control system that would be accessed via a mobile phone application, or on a 5” touchscreen display which would be located in the side doors in the back of the car.



Advantages:
+ Intuitive design that most people will understand
+ Ability to select from multiple libraries
+ Each person has their own screen, so people don’t fight over control

Disadvantages:
- May be unsuitable for people with travel sickness

Concept 2 - Technical details

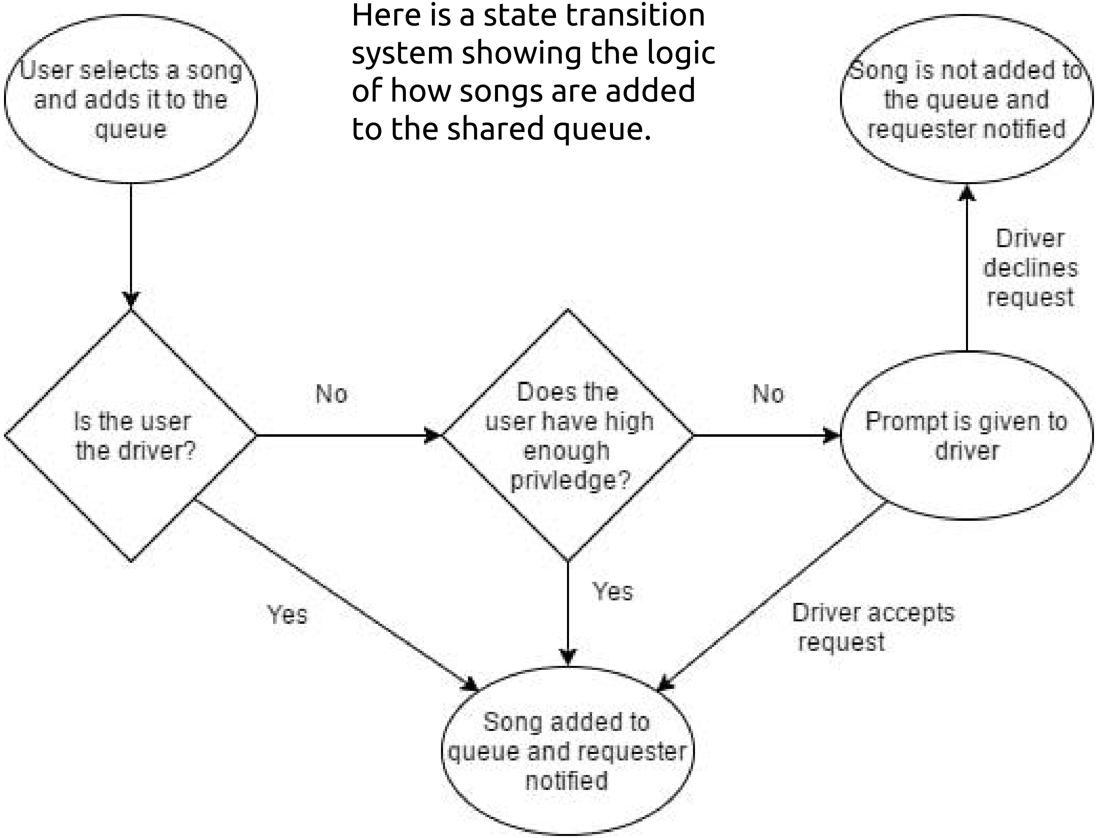
We created state transition diagrams for some aspects of the permissions system for the shared queue concept.
These state transition diagrams should be able to be handed straight to developers to have features implemented.

Devices connect through standard cabling in the car such as ethernet, wireless would add unnecessary complications.

Messages would be sent through the system in the form of JSON, this allows us to structure the data easily in a lightweight, fast and easy to use format.

The user system is stored in the cloud and any changes made are synchronised with the server.

Here is a state transition system showing the logic of how songs are added to the shared queue.



Final design

For our final design we decided to proceed with concept 2 but include some ideas from concept 1 such as voice control because it was a highly demanded feature during our ethnographic research, conducted on a sample of over 70 participants.

We decided not to continue with individual sound system for each user because we believe the queue system was easy to expand upon, implement and it encourages extraversion between passengers on car journeys.

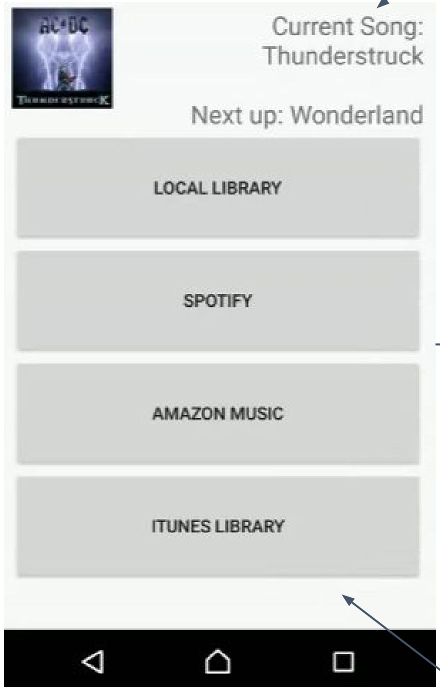
This design improves on existing systems by providing a music system that brings together multiple platforms (Spotify, iTunes etc) into one. It also allows groups to build playlists collaboratively with the option to restrict some people such as children.

Final version

We have created a prototype app to model how the screens in the back doors might look and work, with limited functionality.
A recording of this app prototype in action is available here:
https://gfycat.com/TastyCharmingAnole

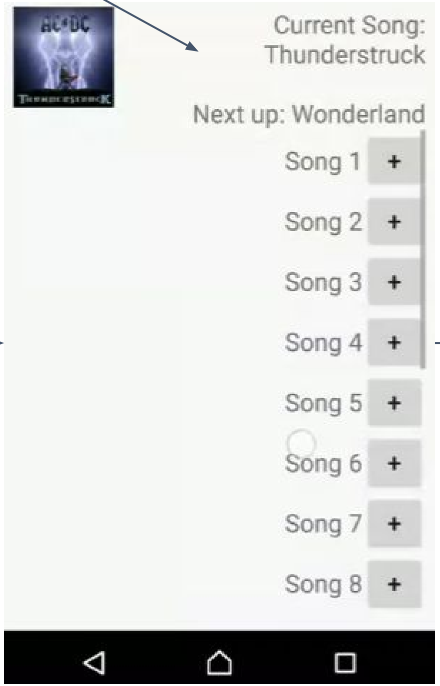
Playlist information retained at top of screen

Main Menu



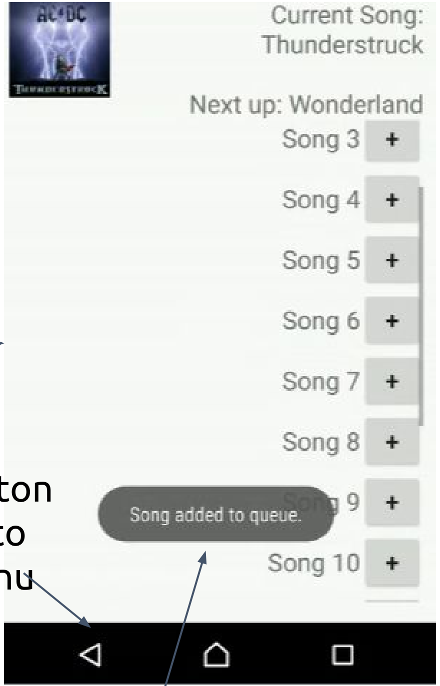
Prompts to choose a library to see music from

Song chooser



Intuitive sliding selection panel for song selection

Song selection



'Back' button returns to main menu

Toast messages for feedback to the user

Final design

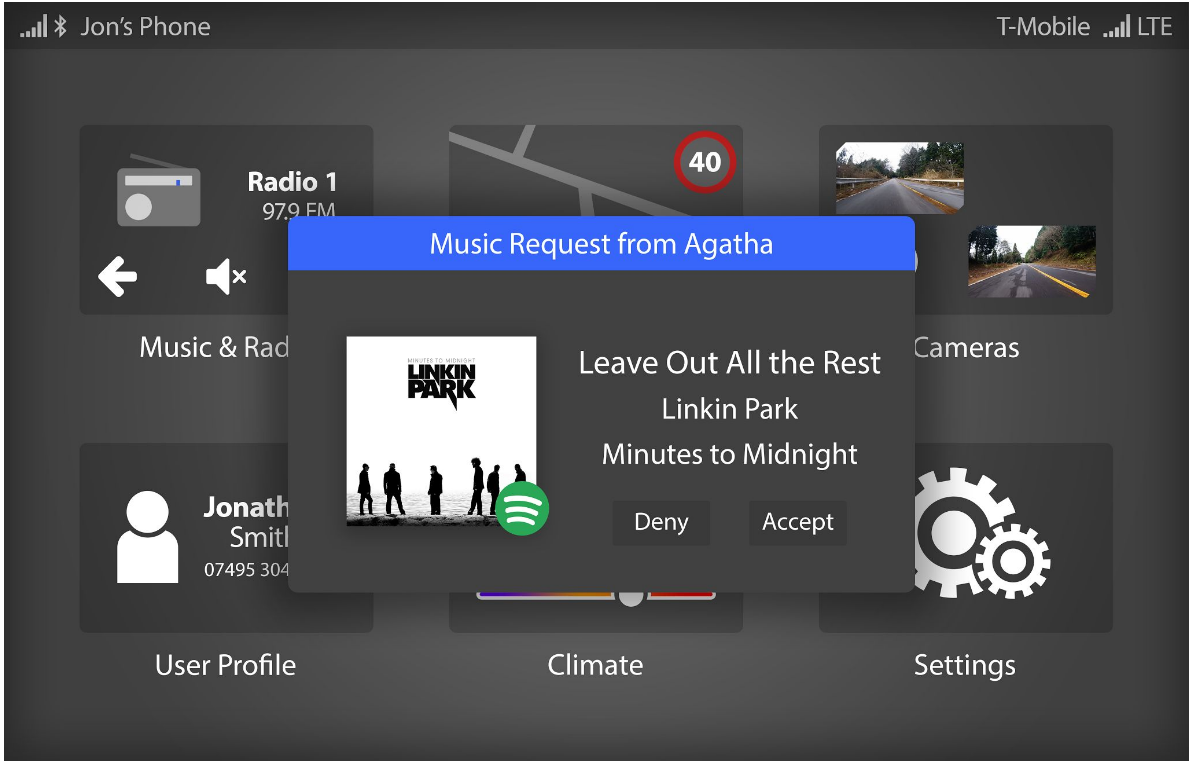
Messages are sent to and from the centre console over wifi or bluetooth on mobile devices. Ethernet will be used for the devices installed in the doors.

A message consists of a unique identifier for the requesting user, the song requested and the library where the song is located (E.G. Youtube, Spotify, Local Library, etc.).

After the console receives the message, the logic described in concept 2 is applied and the sender is given a message describing the action that took place, either that the song has been added to queue or declined.

To prevent users from spamming the centre console with messages, which could potentially distract the driver, each queue addition from a user has to be accepted before another request can be sent.

Music request interface for the centre console



Comments

We think that we have improved our process after the first assignment, taking into account personas, online questionnaires and general planning structures. For example, by switching from creating questionnaires for people to fill out on paper and getting a small range of answers, to using online questionnaires, we increased to receiving over 70 responses.

We attempted to come up with innovative ideas to make our system stand out from the rest and generally improve user experience, unfortunately we struggled to separate ideas into different concepts. We found it hard to split up ideas because when we imagined integrating them into the car, they quite often would all be able to work together as one system.

We knew that for this task, more specification was required on the technical details of our system such as security. We found it hard to include all of the details that we had thought about while still sticking to the rather restrictive template slides. We have created more technical software designs but had to limit ourselves with what was included in our submission.

We enjoyed the task, being able to fix our own frustrations with existing systems in our own designs. The queuing abilities especially excite us since car trips end up often limited to one person controlling the music.

Navigation

The A-Team

Brainstorm

After reading the task, we attempted to note down all the problems that we could think of with current Sat Nav systems. We were able to come up with a lot of problems that we think we could improve in our own system.

We considered the personas that we had created in the previous task and attempted to improve and create ideas to suit each person.

Once we knew what we wanted to improve, we further understood the problem and were able to start planning our brainstorming and generating ideas for QOC analysis.

We came up with numerous ideas to improve various aspects of a Sat Nav. Overall, the brainstorming session went well, resulting in many ideas being generated, meaning that we were ready to go into more detail and refine some of them.

Personas: bit.ly/2kZQGLS

Updates

- Over-the-air updates (wifi)
- Backup system loaded if the system fails. Allows for resetting the system OS, downloading the latest version and trying again.
- Can be updated offline via SD card (**Irene**)

Alternative routes with their associated time to destination

- Few fastest routes (similar to Google maps)
- Provide options to the driver
  - More scenic (**Emily**)
    - Provide routes where roads are more spaced out
  - No tolls
  - No motorways (**Emily**)
  - Slower speed (**Irene**)
  - More economic
    - Long roads low speed limit (Possibly slightly longer than original route)
- Have preferences driver can save and automatically select them
- If drivers are tired maybe suggest routes with a lot of turns to keep driver awake
- Take into account if there is a large group of drivers going in one direction (Drivers leaving a football game) (**Chad, Tony, Paul**)
- Suggest park and ride if used often

Convoys

- Track other drivers in your convoy via GPS (**Paul**)
  - Group announcements / group calls
  - Notify others of the location where someone else stopped
- Synchronised music/playlist (**Paul**)
- Custom pins on maps (animated? coloured?)
- Notifications of members reaching waypoints
- Route sharing Pre-forming route prior to journey
  - On the fly updating
  - Stop off point scheduling
- How to invite drivers
  - Shared key
    - Can't be guessed
    - Easy to input

Brainstorm summary

Criteria:

- Global
  - Clear
  - Ease of use
    - With disabilities
- Routing
  - Time efficient
  - Computationally efficient
  - Realistic
  - Reliable
  - Compliant with drivers habits & preferences
- Interface
  - Not cluttered
  - Aesthetically pleasing (Pretty subjective? Perhaps customisable in colour and style)
  - Easily accessible (For disabled individuals)
  -
- Directions
  - Informative
  - Not distracting
  - Not easy to miss
  - Easy to interpret
- Map Drawing
  - Easy to see your location
  -
- Hardware
  - Not excessively expensive
  - Not easy to break

Once we had completed our initial brainstorming phase, we had plenty of ideas that we needed to group and choose between.

First, we went through all of our initial ideas and ruled out any that we thought may be infeasible.

We looked at the initial problems and ideas that we had written down and came up with questions and criteria to match our ideas into QOC matrices.

The screenshot below shows one of our QOC matrices. We created many matrices that helped us compare different options for the features of the Sat Nav. In this case how we should display the map on a Sat Nav.

How do we display the map?

↓ **Criteria/Options→**	Top-down roads (Birds eye view) Varying Zoom	Top-down satellite	Top-down Zoomed in	Top-down Zoomed out	Head-on 2D (1st/3rd person)	Head-on 3D
Clear at high speeds (H)	y	n	n	y	y	y
Clear on small roads (H)	y	n	y	n	y	n
Looks good (M)	n	y	n	n	n	y
Not (potentially) distracting (H)	y	n	y	y	y	n
Quick to render (M)	y	n	y	y	y	n
Low disk usage (L)	y	n	y	y	y	n
	3H + 2M + 1L	1M	2H + 2M + 1L	2H + 1M + 1L	3H + 1M + 1L	1H + 1M
Score (H=3, M=2, L=1)	12	2	8	9	12	5

Research

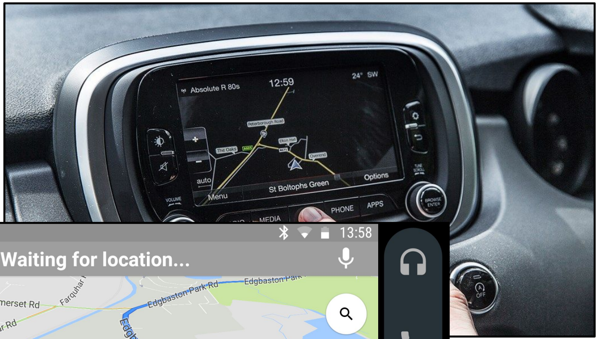
Details of research we've undertaken

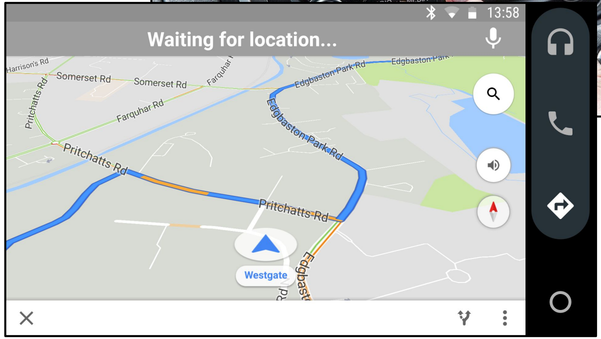
For our descriptive research, we created a questionnaire about usage and opinions on Sat Nav systems. Most of the questions were close-ended for ease of analysis, however, a couple of questions allowed the user to expand on some areas and give suggestions. The questionnaire was shared over social media and received over 40 responses.

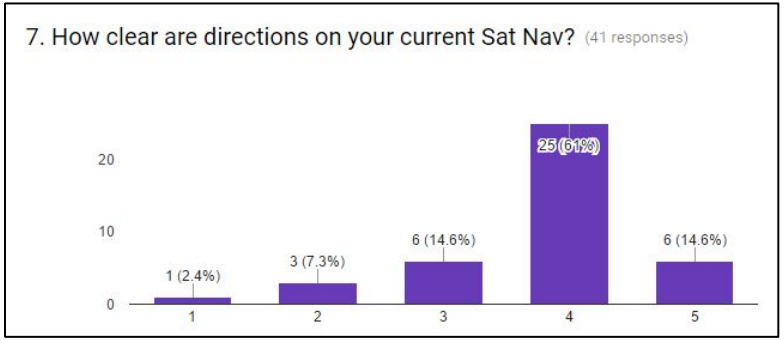
By analysing the results we have extrapolated information that hinted at which areas to focus on. The results illustrated in the image at the bottom right of this slide suggests that current methods of information delivery are perceived well by most users and therefore work well and don't need much improvement.

As part of our external research, we browsed the internet to find out which ideas that emerged from our refined brainstorm had not yet been implemented and which ones could be improved upon. We believe that our most original idea was the "convoy" feature (outlined in design).

Questionnaire results: [bit.ly/2l04Pst](bit.ly/2l04Pst)







Routing - Algorithm

Decision

Our algorithm of choice for computing routes is Bidirectional A*. The choice is based on:

- Computational performance when compared to existing approaches with the same purpose. [1]
- Flexibility in terms of adding different heuristics so that we can compute multiple routes based on route requirements, learning user's habits and preferences, roadwork and traffic information and other such criteria.
- The big advantage of using a bidirectional algorithm is that if you take a wrong turn, you can search again from a new start point. The backwards search tree (already stored in memory upon route selection) is unchanged since your destination did not change.

Search

We are representing the map as a set of nodes and edges where junctions, start position and destination are nodes and the edges are roads between these nodes.

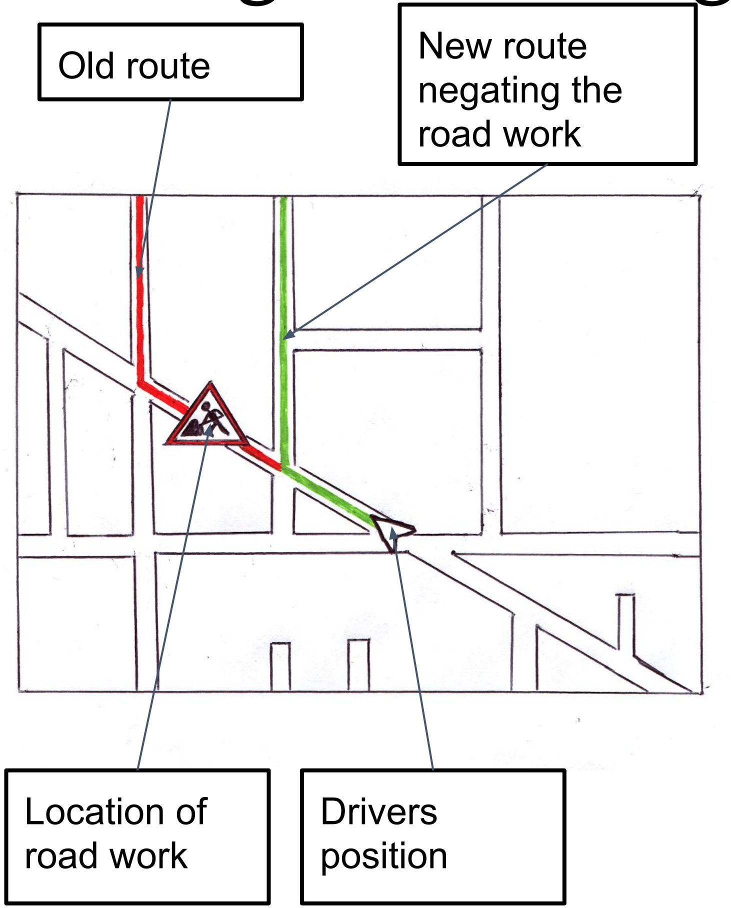
The search can be broken down into groups of road size e.g. (M, A and B roads) so that firstly larger road routes are calculated first such as M roads. From this, we build a route using A roads and so on until we reach our destination from the start position.

Congestion avoidance:

- When a busy venue is to be vacated (e.g. After a big sporting event) or for leaving city centre at rush hour.
- Efficiently distribute cars on separate routes using statistical data with the aim to average out the amount of delay on each road.
- There are opportunities to work with local councils to provide additional useful data and planning.

[1] This is based on tests that we have conducted ourselves, later confirmed by benchmark data.

Routing - Re-routing



Our system will pull in live traffic and road work data from external sources and APIs such as https://developer.here.com. The traffic sources will be polled every 5 minutes to find any accidents, roadworks, closed roads etc. that are on the route of the drive. If an issue is found on the current route then the system will run its routing algorithm again, excluding the affected roads, if a faster route is found then a prompt will appear to the driver to accept the faster route. To reduce the chance of distracting the driver, the driver will also be able to set faster routes to be accepted automatically.

If the driver takes a wrong turn then the system will need to adjust its route. There are two different methods of re-routing the user that had to be considered because of some of the features in our system. The typical method for re-routing the user would be to recalculate the whole route to find the fastest way to get to the destination. In the case of cars in convoy, they will need to rejoin the route that other members in the convoy are following. To accomplish this, the only part of the route that will change is the current part of the journey. The system will change the current route to take the fastest route to an existing waypoint or closest node in the existing route (nodes representing junctions in our map).

Routing - From external devices

Our system will allow the user to share routes and destination searches they have made on their mobile phone with their car software system.

Mobile Phone App

We plan to have a smartphone application where users can create routes that can be added to their car navigation system. This data will be sent to the cloud and then downloaded to the car, users can then select these destinations in the car system.

Users also have the option of sending the routes to the car system using Bluetooth since their mobile phones will likely be connected anyway.

With this application, we could ask for permissions to access internet search history and from this, we can find locations users have recently been searching. We could suggest these locations to the user on their car system when they are entering a destination.

There are privacy concerns with this feature, since users may not want our app to access their browsing history, for this reason, the feature is only optional.

This mobile application could also be used to set up and organise convoy groups using the data from the phone like contact details & pictures for profile images.

Routing - Learning

Preferences

In our implementation of the UI we plan to let the user select some route preferences including, but not limited to:

- Fastest route
- Economical route
- Free-flowing route (still keeping journey length reasonable)
- Scenic route

Contextually Aware Routing

- The routing algorithm will have the ability to learn from the actions of the driver in situations such as those where the suggested route is poor and the driver takes a faster, shorter or otherwise more optimal route to their destination.
  This is particularly enforced if the route taken is one to a frequent destination or on a daily commute. If a diversion is used on a semi-regular basis then the algorithm can learn and use that route on later occasions.
- There will always be an option to override the default routing algorithm and force a ‘fastest route’, ignoring all of the learned preferences. Additionally, an option to ‘forget’ the learnt route.

Routing suggestions

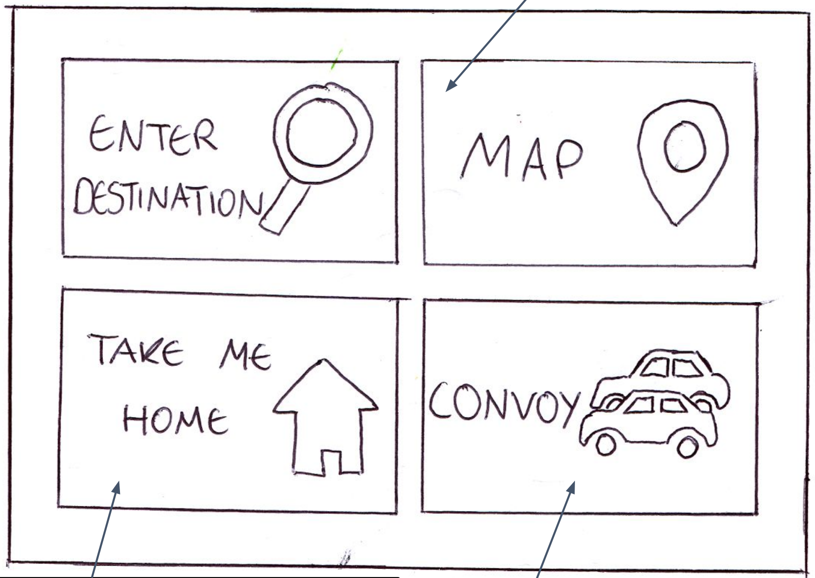
Our system will intelligently create suggestions for destinations based on data collected from many relevant sources including, but not limited to:

- Recent destination search history in Maps, both in the car and on any linked phone app
- Locations of calendar events based on either globally recognisable addresses or locally identifiable locations.
- Addresses of recently searched or contacted phone contacts: home, work or otherwise.
  - These are also likely to show up if any contacts are attending the same events (found from calendars & social media).
- Nearby business suggestions are taken from local mapping APIs like Google or Apple Maps
- Context-aware specific destinations based on already visited locations (see left), time of day, company & passengers etc.

Interface Design

Sat Nav Main Menu

Easy to press large buttons



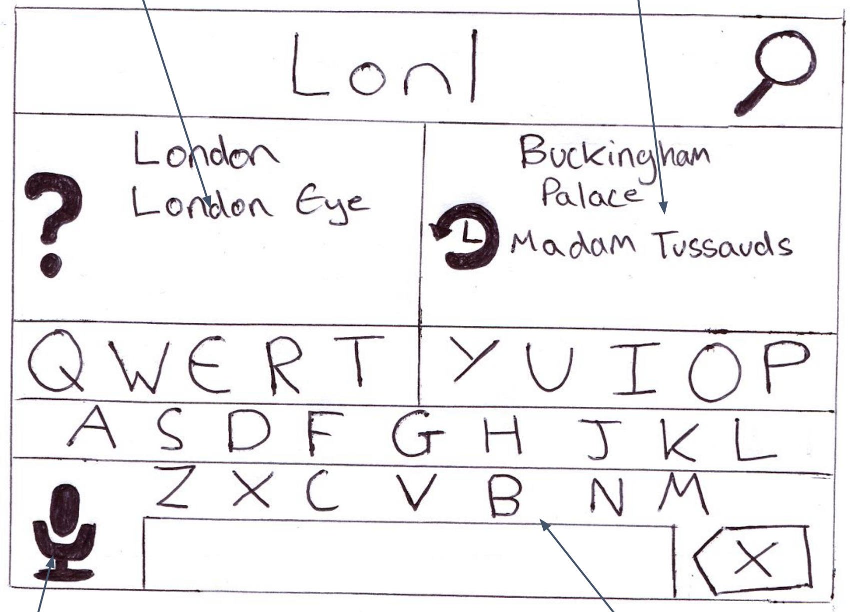
Button to set destination to home with no extra input

Set up or join a convoy

Destination Screen

Suggestions based on search text

Suggestions based on location history & search text



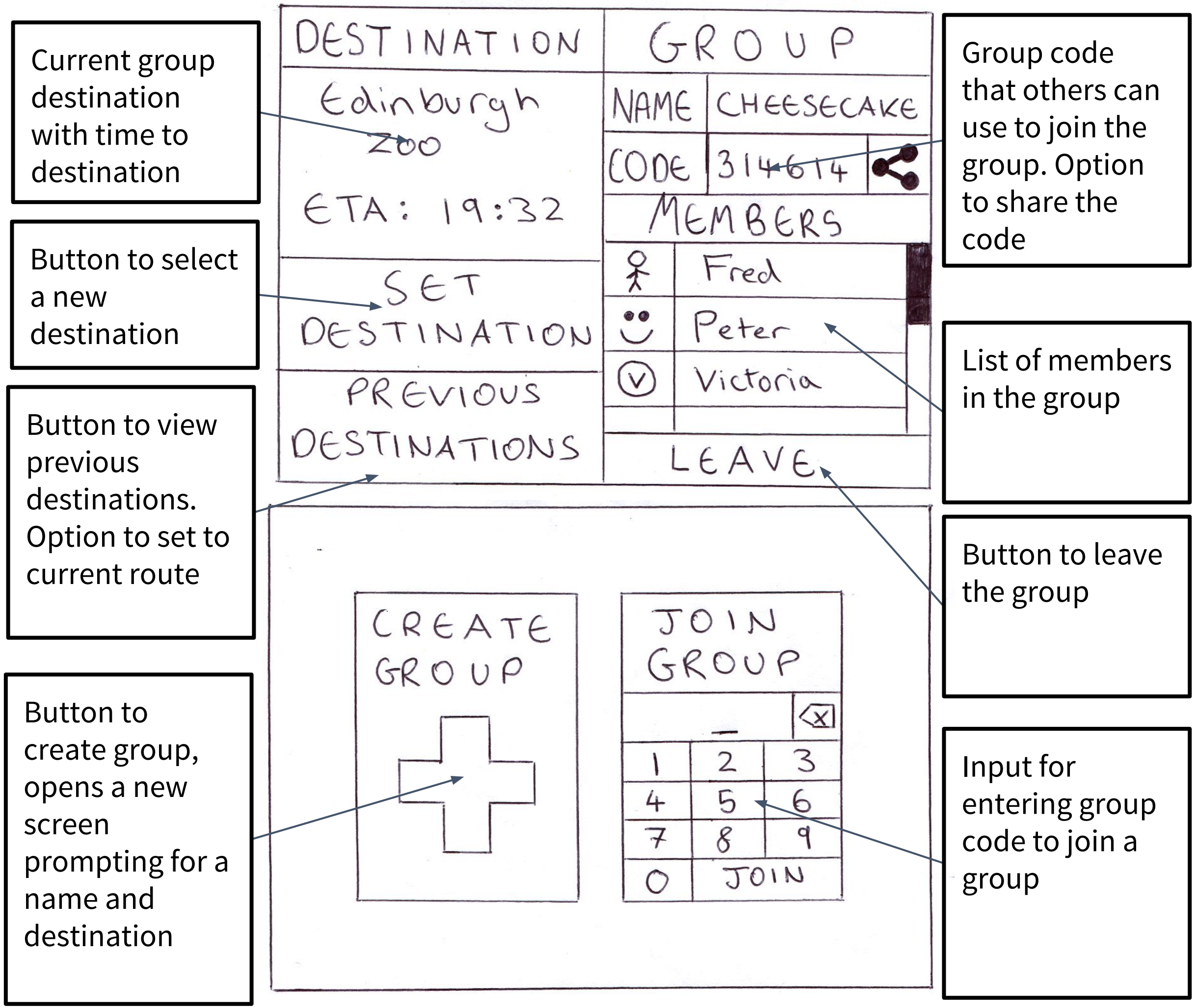
Large mic button to input destination via voice

Touch keyboard with large keys for ease of use

Convoy Tracking

Key features

- Ability to track friends and colleagues when travelling together in several cars
  - Will be able to see other members on the map as circles
  - These circles will be animated to show the state of the member (e.g. stopped, on the wrong route).
- Makes coordinating breaks and detours much easier (e.g. meals, rest, refuelling, etc.)
  - Done via announcements or adding waypoint pins to map
- Ability to share music and other entertainment services provided by the in-car systems.
- Ability to have group voice, video and text chats over the convoy group.



Convoy Tracking 1

Tracking & position sharing

Standard GPS, along with other positioning techniques based on other radios in the car, are used to track the cars; the cars upload their locations to a cloud server, which in turn pings this location to the other cars using secure asymmetric encryption, only decryptable by the members of the convoy. The data will be deleted from the cloud and cars when a new location for each car is received, or when either person leaves the convoy.

Creating & joining a group

- One person will create a convoy group
  - On their phone or on the center console in the car
  - The driver is given a 6 digit code.
    - Others use this code to join the group.

Leaving a group

- Prompts are given to the driver asking them to leave the convoy when:
  - The current driver of the car changes.
  - All members of the group reach the destination.
- There is an automatic timeout where a car leaves the convoy after having been off for 24 hours.

Routing option 1: Shared route

- Designed for when the cars start fairly close together, the routes merge onto one central route with shared waypoints.
- Creator of group generates their individual route, the other cars then route to the nearest waypoint on that master route.
  - Can utilise K-means clustering, detailed in a later slide.

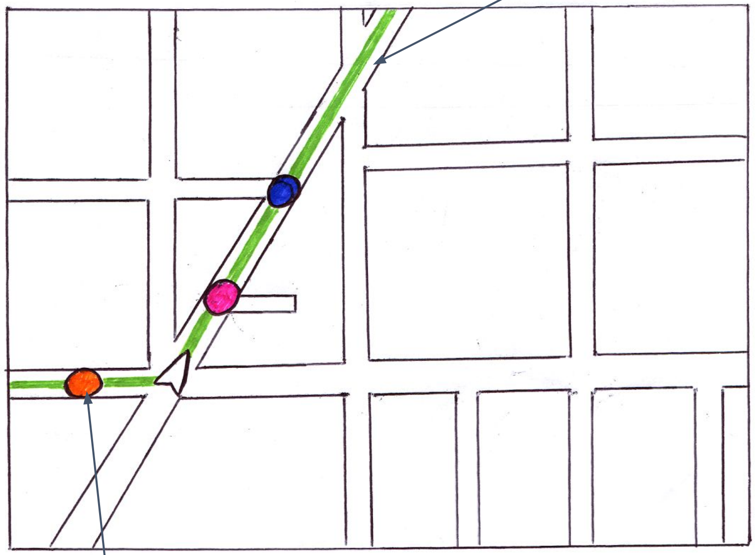
Routing option 2: Individual routes

- Designed for when the cars start far apart and the participants just want to reach the same destination while knowing how far away the others are.
- Routes are calculated separately on each car’s local system.
- May end up on the same route eventually.
- Each machine uploads the route to the cloud, the system checks if routes become identical and if they are, merges these users onto a shared route.

Convoy Tracking 2

Shared route

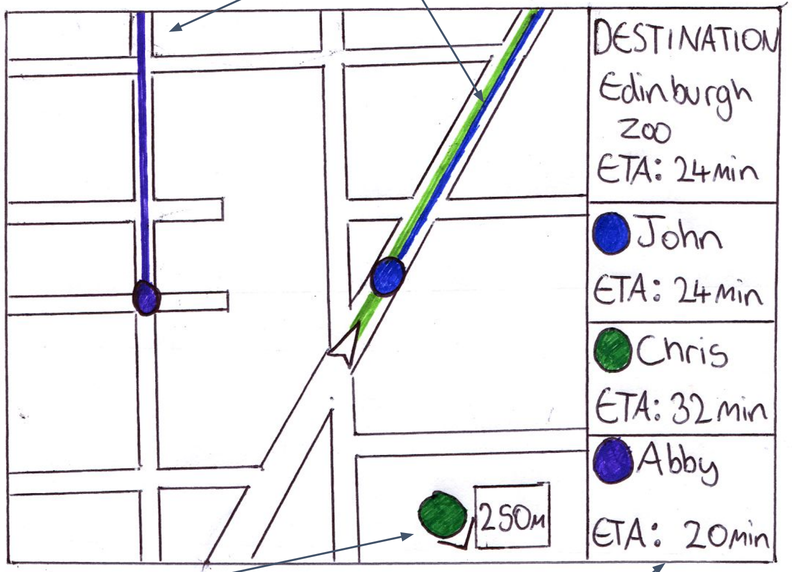
Everyone follows the same route



Each car has its own icon

Individual routes

Everyone has their own route

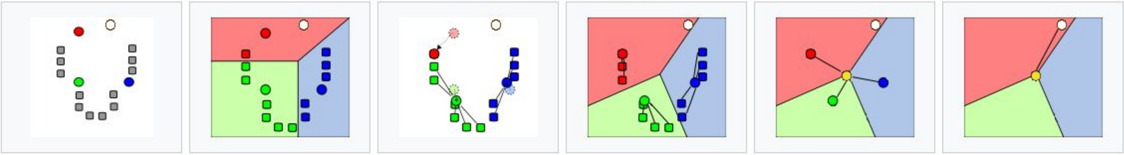


Shows direction of people who are nearby but off the map (this is hidden if they are very far away)

Shows each person’s ETA on the side, since they won’t always be visible on the map

Convoy Tracking 3

Demonstration of the standard algorithm



1. k initial "means" (in this case k=3) are randomly generated within the data domain (shown in colour). The grey squares represent car starting nodes, with circles being waypoints.

2. k clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.

3. The centroid of each of the k clusters becomes the new mean.

4. Steps 2 and 3 are repeated until convergence has been reached. The first stage of the routing is performed: routing each car to their clusters waypoint.

5. The second stage of routing is performed, routing from the cluster waypoints to the nearest waypoint to the centerpoint of the clusters.

6. The final routing involves calculating the route from this central waypoint to the final destination.

Routing option 1: Shared route

A possibility for shared routing is to employ K-means clustering as a way to group the members together as quickly and efficiently as possible. Utilising K-means clustering we can find waypoints to route cars to, for example, the green section cars would be routed first to the green waypoint, and then routed to a central location where all drivers then go to the destination.

This method for routing may not always be appropriate, in the above example the top blue car is going quite out of its way to reach the destination, while it may make sense for the green sector cars to meet up since all of them are going the same way. Therefore, if a shared route is employed each driver would be presented with the option to meet as soon as possible (through this method) or to join onto the main route similar to how it was discussed on the previous slide.

On the fly route updating

Any changes made to the current route by the convoy leader will be sent to the cloud and pushed to all convoy members, updating their route. Drivers will be notified when the route has changed. If the destination has changed, the route is recalculated depending on which routing option is used.

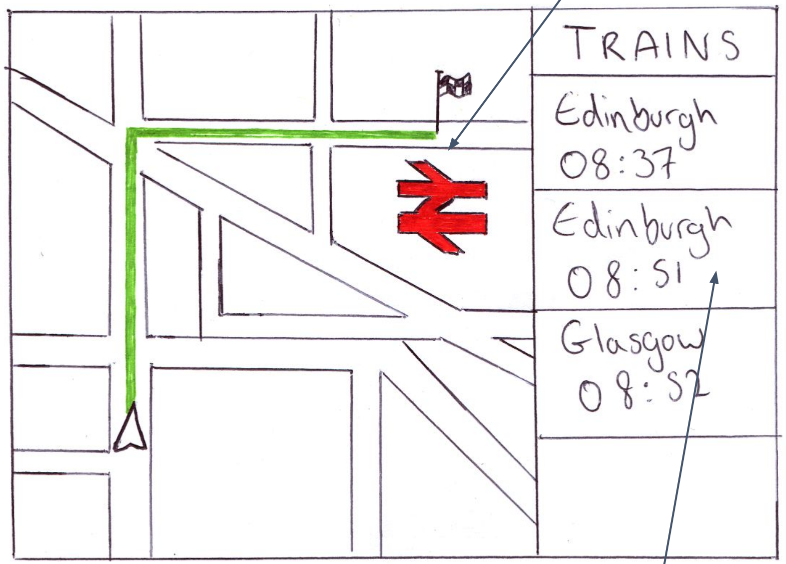
On loss of connection

When a member of the convoy loses internet connection, they will still have the route on their Sat Nav. However, other members of the convoy will not see the location updates of the member losing connection. They will be given a notification and the pin representing the member losing connection will stay in place and fade out until the connection is regained. On reconnection, there is a check to see if the route has changed.

Image adapted from https://en.wikipedia.org/wiki/K-means_clustering

Public Transport

Destination is a train station



List of trains departing within ±15 minutes of ETA

General Explanation

With environmental issues being a widespread concern, we would like to reflect such concerns in our system. This will be done by facilitating (and implicitly encouraging) the use of public transport where possible. We will have a public transport widget, which will automatically show transport departing ±15 minutes from the ETA when the destination is a train, bus or coach station.

As a second subtle push towards public transport usage, our navigation system will suggest drivers aiming for city centres and other popular destinations to make use of one of the many '*Park and Ride*' services which would save the driver time, fuel and would drastically reduce emission (since accelerating and stopping repeatedly is one of the most polluting driving behaviour).

System Updates & The Cloud

Updates

Updates to the car software will primarily be installed *over the air* via recognised, secure WiFi networks or optionally a cellular data connection as enabled by a settings option. Offline updates are also available through the use of a USB storage device either as a fallback method or in case other primary update methods are not available or applicable.

Another supported method for obtaining updates is through the mobile app: updates are downloaded on the driver's smartphone when it is connected to WiFi or another internet connection and then transferred to the car when it is next within range, but not installed immediately. The driver will be notified and asked to confirm a schedule for installing updates at a convenient time.

To ensure the integrity and reduce corruption of the in-car system, all software updates after being downloaded will be verified with MD5/SHA1 to ensure that the file is not corrupted.

Failure

Due to the concern of a possible software failure after updating, we have decided to implement a dual-system architecture, similar to that used in Android. Initially, both systems are flashed with the same image; as updates are installed, the other system image is written to then booted into as the new primary system after a successful install. This leaves the other system as a complete bootable image in the case of corruption or other fatal disasters rendering the updated system unusable. This alternating behaviour of 'ping-pong' between the two partitions is used for every update to ensure a more reliable system.

In the case of a system failure, the corrupt partition is updated with the last known good system image before it is used again.

Cloud System

Notifications and requests are sent to and from the cloud servers in the form of JSON files sent over TLS. As an example, an update to a convoy route would contain data about what the notification is so that it can be handled accordingly by the car and a link to the new routing data for the car to download.

Final version

We decided based on feedback from the questionnaire that we would need a display that doesn't require the driver to look far away from the road. This is to reduce the time a driver isn't concentrated on the road. We decided to have a seperate, smaller display on the dashboard, above the centre console, that is angled towards the driver. This smaller display will have limited functions compared to the centre console display.

We would need the car to have a modern multi-core processor in order to be able to run many of the routing tasks simultaneously and for the system to be able to calculate routes quickly. Since our routing algorithm will be working in stages and not load the whole country and every possible route, we will not need much RAM for the Sat Nav system. Typical Sat Navs have up to 256MB of RAM available. Since our system will be built into the same computer as the media systems, we will likely have about 8GB total memory. Memory is cheap and this 8GB should be plenty for our system.

Through our questionnaire results and research, we've discovered that current in-built Sat Nav systems often have rather poor touchscreen displays, resulting in screen navigation and keyboards being hard to use. We would like our in-built system to have high quality touchscreen displays to make entering destinations using the on-screen keyboard much easier, as if using a modern smartphone. To combat difficult data entry, the user will be able to send locations to the Sat Nav utilising a phone app.

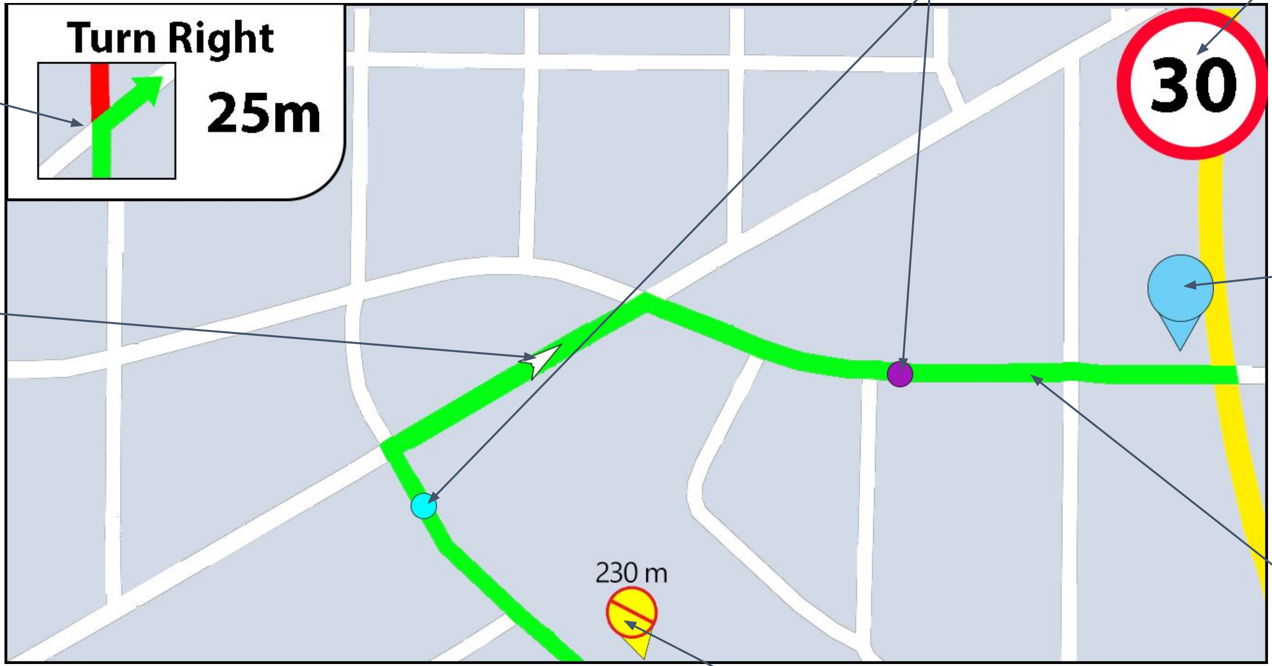
Final design - Convoy

Members of the convoy that are close to the driver

Current speed limit of the road the driver is on

Current directions are displayed. This will be accompanied by a vocal direction

Driver location



Designated stop-off point on the route to the destination

Route of the convoy to the destination. It is shared between all drivers

**Note that our design shows a static map for ease of demonstration purposes, but the implemented design would likely feature a rotating map as per usual Sat Nav design.

For an animation click here: http://imgur.com/a/xck0U (Annotated with yellow boxes)

A member of the convoy who is far away and has come to an unexpected stop. The location of the member is described by the direction of the arrow

Comments

Our design process this week was rather different. After the lecture where we learnt how to use QOC analysis, we decided to use it in our design process to make decisions on the best way to implement our ideas. Using the QOC matrices felt very different and unfamiliar to how we'd done things before, we had a bit of trouble initially because our questions were not specific enough. Once we got used to using the technique, we found it very useful to determine pros and cons of feature implementations.

Since we had limited time and space on the slides and we wanted to go into detail on the ideas we had, we decided not to go ahead with some initial ideas such as guided tours. While we felt that they may be good, other features were much more inviting and potentially useful.

As we explored some ideas, we came to the conclusion that they needed to be split up into smaller parts for different user's uses. For example, we initially wanted a convoy feature that would allow for drivers to follow the same route, but what happens if they want to leave from different locations? This is one of the ideas that had to be split by clear implementation differences.

The whole task was a tricky one to innovate in besides what had already been mentioned in the assignment brief. Satellite navigation systems are used all the time and have been around for a long time. While car Sat Nav systems do seem to lag behind, creating new ideas that aren't already found elsewhere is a hard task.

Car Safety

A-Team

Brainstorm

After learning the task was based on car safety we immediately started noting down some of our perceived problems with car safety, and all of our ideas to counteract the leading causes of car accidents.

At times, we struggled to be able to think of new, innovative ways to improve car safety systems. By coming up with and noting down even absurd ideas, we sometimes were able to inspire more practical ideas.

We also used lateral thinking in order to come up with new ideas. For example, we tried to think how to keep people in the car safe without the typical airbags. We started looking into airbags built into the seat that could wrap around the user.

A successful brainstorming session gave us plenty of ideas to build upon for the rest of our work.

Initial Brainstorm: https://bit.ly/2mvkPE7

Tired

- Sound
  - Louder music
  - Annoying sounds like an alarm (Periodically perhaps)
- Lighting
  - (Annoying) flashing lights
  - Light up inside car
    - Standard in-car lighting
    - Strobe lighting / disco lighting
- Massage chair
- Jerk steering controls (what do we do in events where this is required?)
- Vibrate stuff (seat, steering wheel)
- Enable autopilot/lane assist
- Detection
  - Lower heart rate
  - Camera, watching eyes
    - Rubbing eyes
    - Speed of blink
    - Head pose
  - Watch speed fluctuation or veering
  - Yawning
- Turn on hazard lights
- Suggest breaks every 2 hours or if tiredness detected.
  - Suggest route directly to nearest service station
- In-built coffee machine/energy drink dispenser (Something that can be made of highly concentrated powder and is not difficult & time consuming to drink… Like a shot)
- Temperature control
  - A/C
  - Open window
  - Fluctuate between warm and cold

Alcohol-related

- Detection
  - Erratic driving
    - Swerving a lot
  - Slow reaction times
  - Large pupils
  - Breathalyser
- Solution

Brainstorm summary

After forming our initial ideas, we went through each point and discussed how to improve/expand on them, and removed some of the more crazy ideas by not moving them into this document. We also began research on some points in order to gain more of an insight into the current technology, and for a proof of concept for certain things.

After this expansion, we marked some ideas to not discuss further (highlighted in red) with reasonings highlighted in orange, these were ideas that we thought might be unsafe/impractical to incorporate into the car's safety features.

After summarising our brainstorm we had plenty of research, details and understanding of how our safety ideas would come together to form a full system.

Tiredness

- Detection
  - Lower pulse (sensors in steering wheel)
    - Redundant, since we can get enough information from cameras
  - Camera, watching eyes
    - Located above sun visor
    - Rubbing eyes
    - Speed of blink
    - Privacy concerns, especially from our Irene persona
  - Head pose http://breckon.eu/toby/publications/papers/walger14headpose.pdf
  - Watch speed fluctuation or veering
    - Camera facing forward watching the lines of lanes
  - Yawning
- Solution
  - Sound
    - Louder music
    - Annoying sounds like an alarm (Periodically perhaps)
    - A new song is played
  - Lighting
    - (Annoying) flashing lights
      - Not good to do this in case of medical conditions / risk of annoying the driver too much
      - Distracting to other drivers
    - Light up inside car
      - Standard in-car lighting
      - Causes visibility issues due to contrast in lighting
      - Distracting to other drivers
  - Vibrate stuff (seat, steering wheel)
    - Not sensible
  - Enable autopilot/lane assist
  - Suggest breaks every 2 hours or if tiredness detected.
    - Suggest route directly to nearest service station
  - Temperature control
    - A/C
    - Open window
      - A/C will be more effective

Refined Ideas: https://bit.ly/2lg3Vax

Research

A large part of the research for this topic involved the causes of traffic accidents since this would be one of the largest safety concerns we could target in our designs. [1]We used an online list of causes of accidents to ensure that we have at least discussed each point; categorising most of them as bad habits to be included in our bad habit prevention such as speeding, unsafe lane changes and tailgating. We supported this list with further research on how dangerous it is to drive while tired, finding that in a 2005 study, “more than one-third [of Americans], (37% or 103 million people), have actually fallen asleep at the wheel”[2] which told us that battling tiredness is a large requirement for this task.

We carried out a questionnaire[3] in order to find out more information that would help us rank the importance of some of our safety ideas. The questionnaire also gave the opportunity for users to provide feedback on what they would like in a car’s safety system. For example, one response said that it would be a good idea to reward good driving in some way and have a warning light for when you are driving too close to the car in front. The questionnaire also verified our views that people’s driving is negatively affected by road rage and that people don’t take the recommended breaks while driving to prevent tiredness.

[1] https://seriousaccidents.com/legal-advice/top-causes-of-car-accidents/
[2] http://drowsydriving.org/about/facts-and-stats/
[3] https://bit.ly/2lgmf3c

Battling Tiredness - Detection

Detection

One of the leading causes of accidents is drowsiness. In our system we will attempt to detect if the driver is fatigued by having a built-in camera located above the sun visor, this will be focused on the driver's face. With this camera, we will monitor the driver's head pose (more detail on a later slide), how slowly they are blinking and actions such as yawning or rubbing their eyes.

If drivers are constantly dropping their head to the side or downwards and performing actions related to fatigue there is evidence that they are tired. We can combine this information with data on the car itself, such as speed fluctuations or if the driver is veering to further back up this evidence. We can detect the driver is veering out of their lane with a front facing camera using lane detection[1].

Obviously, there are privacy issues with a camera watching the driver all of the time they are driving and so the detection by the camera can be disabled by the user. This will result in the system relying only on the other sensors (lane veering and speed fluctuation) to detect tiredness.

[1] https://www.vision.caltech.edu/malaa/publications/aly08realtime.pdf

Battling Tiredness - Solution

Solution

After detecting that the driver is fatigued, the driver will be notified that they are tired and will be suggested to take a break from driving. The car can assist in keeping the driver awake and alert until they take a break using the following methods:

One method is to stimulate the driver by playing music louder, playing sounds like an alarm possibly even periodically if the driver is showing more signs of fatigue.

Another method is to change the temperature of the car using the air conditioning to reduce the car's temperature since this will keep the driver more alert[1].

In order to assist fatigued drivers, there is an option to enable lane assistance, which will aid the driver by counter-steering cars back in lane if they begin to drift out.

The previously mentioned methods of keeping drivers alert are only a temporary solution until the driver takes a break. We will suggest that the driver takes a break from driving every 2 hours or if they seem tired and even suggest a route to the nearest service station.

We also thought about other solutions such as having bright lights flash to keep the driver awake but this has the issue of distracting other drivers on the road especially when it is dark. It also has problems if the driver has a medical condition and in general, it will probably annoy the driver.

[1] https://www.scientificamerican.com/article/warm-weather-makes-it-hard-think-straight/

Distractions

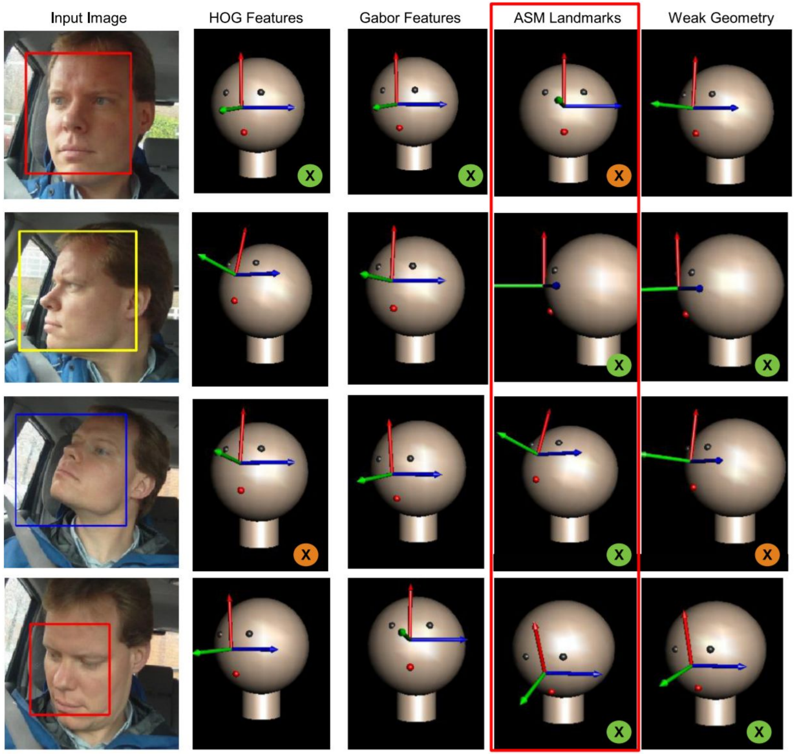


Fig. 4. Head pose estimation under varying non-uniform illumination conditions

As mentioned in the tiredness section, we have a camera facing the driver to detect certain features such as head pose, and blinking rate/length. To extend the head pose detection, we will be utilising ASM landmark detection[1] to determine where the driver is looking. With this model, we will know whether the driver is facing forward or elsewhere. Due to distractions taking place over longer lengths of time, a small history will be kept (such as over the past minute) in order to model how distracted the driver appears over time. The car will utilise this data to see if the driver is rubber necking (i.e. row 2/3 of fig. 4 as well as the driving speed lowering) or checking their phone (such as in row 4 of fig. 4).

This knowledge can be paired with current context, enforcing that the driver to be more focused on the road if they are moving at 70 mph compared to if they are moving in traffic (though the system will not tolerate much distraction unless the car is stopped).

If the system detects a high amount of distraction, it will announce this with an audible warning telling the to stay focused on the road.

[1]http://breckon.eu/toby/publications/papers/walger14headpose.pdf

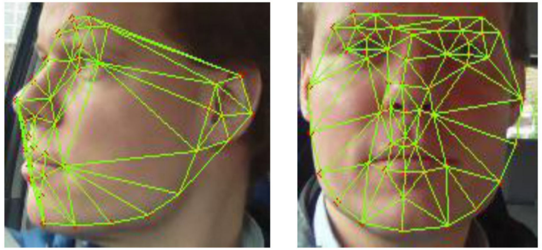


Fig. 1. Active Shape Models for frontal (left) and side profile (right) facial views.

Notifications / Distractions

Distractions relating to mobile phone and other digital devices such as navigation systems, are the cause of over a quarter of all car crashes in America[1] and similar figures follow in the rest of the world. In order to minimise the amount of distraction and therefore reducing the risk of accidents we would like to offer the following options:

- Completely hand over phone management to the in-car infotainment system which will filter through phone notifications and only notify the driver of important notifications (e.g. texts and calls) using appropriate methodologies such as:
    - Speech-to-text
    - Display notifications on instrument cluster for minimal look-away time

- Unimportant notifications such as event reminders, invitations, comments, app reminders, update notifications, etc. will not be displayed at all and will be available to the driver once the car is safely parked with the handbrake (or e-brake) on.

- Driver interaction with navigation systems and mobile phone activities will be possible via dictation while driving and manually while the car is safely parked, as noted above and with the engine off.

- Offer an opt-in do-not-disturb mode especially for times when users are driving; this will send a preset response to contacts that try to call or text while driving. In addition, the driver will be able to list some contacts as exceptions (spouse and children for instance) so that their calls still get through.

[1] http://uk.businessinsider.com/cell-phones-causing-car-crashes-and-deaths-2015-5?r=US&IR=T

Alcohol Related

Driving under the influence of alcohol is at the root of a large proportion of accidents, injuries and deaths. Reports say that in the USA in 2014 almost a third of all traffic-related deaths were caused by impaired driving due to alcohol use[1]. Similar figures are seen in many European countries and therefore we found the urge to address this issue in our design.

The idea: A wired breathalyser will be installed in the driver side door. The driver will then be able to check their alcohol level if they wish to do so since this is hard to gauge, being dependent upon body build, age, metabolism and other factors. If the driver passes the test, the system will congratulate them and wish them a safe drive. On the other hand, if the test fails, a warning will appear on the instrument cluster screen complimented by the values from the test on the centre console screen. In addition, to sensitise the driver to the many risks of driving under the influence, some statistics will be displayed in regards to accidents, injuries and deaths caused by drink-driving. The idea of this follows that of printing graphic warning labels on cigarette packets[2].

Some effects of intoxication will also be picked up by our tiredness detection, such as lane veering. Since our solutions for battling tiredness are also relevant to drunk driving, with the end result being to get the driver safely off the road, this is not an issue and we can utilise this side-effect positively.

[1] https://www.cdc.gov/motorvehiclesafety/impaired_driving/impaired-drv_factsheet.html
[2] https://en.wikipedia.org/wiki/Tobacco_packaging_warning_messages

Bad Driving Habits

As mentioned on the research slide, a large portion of traffic accidents are caused by what we categorise as bad driving habits. In order to combat this, we have come up with a system which will suggest driving improvements to the driver after they commit bad driving habits.

Example of bad driving habits are:

- Speeding
- Running red lights
- Unsafe lane changes
- Wrong way driving
- Poor indicator usage
- Not driving at a constant speed
- Tailgating
- U-turning on roads you shouldn't
- Street racing
- Unnecessary sudden breaking
- Not checking mirrors before indicating

These bad habits will be detected mostly by information stored on the vehicle such as speed, whether or not they have used indicators before changing lanes, the sharpness of breaking compared to objects ahead etc.

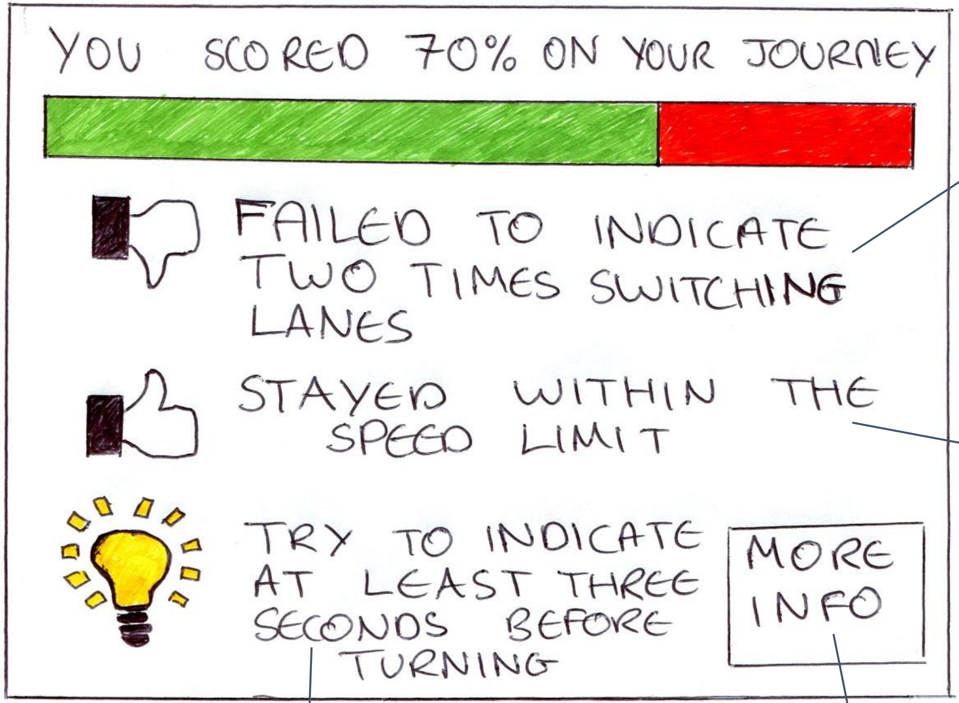
Detection will depend heavily on the sensors in the car and the techniques used to analyse detecting safety elements mentioned in previous slides such as lane drifting.

At the end of a journey, users will be given a summary of their journey highlighting errors they have made, suggesting the user to not do these in the future. This information could also be passed onto insurance companies in order to calculate insurance prices. Which will, in turn, affect the way drivers driver positively.

Bad Driving Habits

We will also give each journey a score which will be linked to the user account of the driver (mentioned in previous weeks). This information can be used to build a leaderboard for all drivers of the same car, and even have the possibility to integrate social media to compare driver score with friends and family. This will motivate the average driver to drive in a way that improves their score and therefore their safety while driving.

To the right is an image of an interface that would be displayed to the driver on the centre console after finishing a journey.



Negative driving habit. Brief description of the fault.

Positive driving habit so the user doesn't feel they are always driving badly

Driving improvement suggestion

Button to show more detailed information on faults

Road Rage

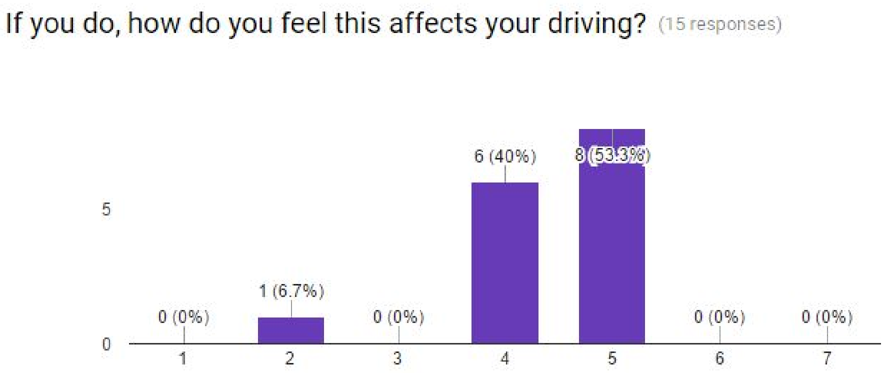
Drivers will often get angry at certain events while driving, which tends to lead to worse driving, as evidenced by our survey[1]. Therefore, we aim to detect when the driver is angry and attempt to calm them down, to reduce the risk of accidents.

Detection:

- Aggressive behaviour, such as hitting the steering wheel or slamming the doors shut.
- Aggressive driving, such as sharp acceleration & braking.
- Swearing - the microphone (used for standard voice commands) will start recording when high volume speech is heard.

Solution:

- Linking back to our music designs, we can start
- a playlist or radio station that the driver finds soothing/relaxing.
- Linking back to our navigation designs, we can re-route away from busier roads or near to scenery the driver finds pleasing.



[1] https://bit.ly/2lgmf3c

Cyclists / Motorcyclists

Problems:

- Hard to spot from high vehicles (HGVs, Buses, Vans, etc.) and even harder to spot at night unless equipped with appropriate lights and high visibility clothing.
- Often requires the average driver to actively be on the lookout for these vehicles in order to spot them, which is uncommon.
- Can creep up on the side of stationary vehicles at traffic lights and often not respect them (bicycles) and cause confusion/accidents if the lights go green shortly after since the car drivers may be unaware of them.

Solutions:

Solutions to the issues outlined above lie in aiding detecting such vehicles and making car drivers aware of their presence. Some suitable methods to do this are:

- Automated recognition using cameras and machine learning. (Train the system with a set of training images containing cyclists and motorcyclists having given the system the location of the feature of interest, later tested on a test sample before deployment)
- Make of electronic tags installed on bicycles and motorbikes, detected by infrared sensors fitted on the sides, front and back of the car[1].
- Both of the methods above will notify the driver once one of the vehicles above is detected using both:
  - Visual warning - A top-down view of the vehicle will be shown on the instrument cluster display along with a red dot marking the approximate position of the biker relative to the car.
  - Sound warning - A series of sound signals will come from the direction where the biker is detected through the surround sound system of the car.

[1] http://www.cyclealert.com/

Road Accidents - Prevention

There will be many small safety features to combat accidents and assist accident-avoidance within the car ecosystem that together will make a safer driving experience.

The car continually maps it's surroundings with cameras, IR and radar sensors around the exterior of the car and repeatedly simulates the near future to predict any potential hazards and attempts to mitigate against them

Tesla use a radar system that we intend to incorporate into our cars in their cars projected from all sides to map out surrounding vehicles and can detect and predict the actions of other car drivers and can even preempt collisions before they happen, as captured from a Tesla: youtu.be/_Kti-9qsLpc

Automatic braking
Brakes will be automatically applied with the knowledge and foresight that surrounding cars are getting too close. The car can even apply emergency brakes before the driver can react potentially avoiding a collision or at the very least reducing impact for all parties involved.

Rubbernecking prevention
The in-car safety systems will try to warn and prevent the driver getting distracted whilst driving, this also includes looking out of the window and not at the road. There has been research done into techniques that can be used to detect the direction that the driver is looking, as explained on the *Distractions* page.

We would incorporate these detection algorithms in and use the information to alert the distracted driver by means of visual alerts projected onto the front/side windows and displayed on all internal screens, including the instrument cluster, centre console and all in-door screens accompanied by audible alerts for those not paying attention.

Emergency alerts
We propose a system that integrates with emergency services to alert the driver of nearby emergency vehicles and to and allow extra time for planning a stop or alternate manoeuvre in response to the change of circumstances.

On-screen alerts can be displayed in the dash or windscreen on the whereabouts and heading of the incoming vehicle.

Road Accidents - Collision Assistance

Industry standard safety features such as automatic airbag deployment are intelligently improved by the car to only deploy to occupied seats and those not in use by minors or those who could be ill affected by the use of an airbag.

An 'emergency-mode' will immediately be activated when a crash is detected by the car. This puts the car into a life-preservation mode in an attempt to assist the passengers in survival to the best of it's abilities.

Assuming the crash is significant enough (e.g. the car rolls or takes a hard hit) the emergency services will automatically be contacted and put onto speakerphone with the assumption that the driver is unconscious or severely injured[1]- the option to cancel this will always be presented to all passengers as a fallback. Medical details stored in the accounts of the car driver and any known passengers will also be sent along to the emergency services.

To prevent unwanted explosions or heat-induced damage to people or cars, a system of heat limiting foam fills the engine compartment of the car as a precautionary measure.

One of the many dangers of driving in the rain or snow is being blinded by the reflections of the headlights from the falling water/snow so we will install 'smart headlights' that use DLP projectors and a predictive system to illuminate between the precipitation and significantly improve visibility in the dark. [2]

To protect the ears of drivers about to experience an imminent crash, the in-car entertainment system will project a constant, just about bearable and not painful, 85dB of static noise to prepare the ears for much louder and potentially more damaging crash sound. This technology, developed by Mercedes, can help to reduce ear damage during a crash.

Every seat in the car is fitted with a basic set of sensors designed to monitor the vital signs of all passengers- this information can be relayed to the emergency services teams to aid a successful rescue from the potential car wreckage .

[1] Emergency services calling was discussed in more detail in the dashboard slides
[2] https://phys.org/news/2012-07-smart-headlights-drivers-raindrops.html
[3] https://www.wired.com/2015/07/mercedes-using-loud-static-protect-fancy-ears-crashes/

Hardware Requirements

Firstly, a key set of hardware we need is the sensors. These have been mentioned on previous slides but in general, we will have cameras to provide 360-degree vision around the car, including a camera facing forward used for lane detection; a camera facing backward for a rear view feed; and a camera focused on the driver's head area (which can be disabled if the driver desires). The car will also have the usual contextual sensors such as accelerometers (for the car in general, as well as the doors and steering wheel as part of our rage detection) and a gyroscope. There will also be the breathalyser, as mentioned earlier, and the microphone for voice commands and swearing/yelling detection. These are here alongside the standard sensors widely available in cars already, such as seatbelt detection/reminders (which we will expand to also detect the passengers[1])

Along with seatbelt detections, we are also including seatbelt pre-tensioners (as standard) to keep the passengers in a secure, safe position during a crash. This will be utilised in further safety features such as a helmet-style airbag, as seen in cyclist technology[2] (or as close to a helmet as possible in order to keep the head secure, since the full-head helmet may not be fully possible from just the headrest) deployed from the headrests, which only activate if the person's head is in the right place (with proximity sensors) - where the pretensioner aims to keep the person against their chair for the headrest airbag to sequentially keep their head secure. A wide array of other airbags will also be installed for further safety, such as for overhead, side, and knees, as used by Tesla.[3]

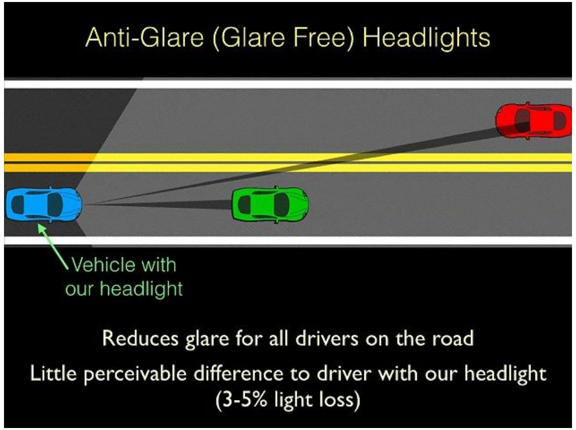
[2] Cyclist helmet formed as an airbag from a scarf

[1] https://www.iee.lu/en/products/automotive/seat-belt-reminder
[3] https://www.tesla.com/support/model-x-specifications (safety section)

Hardware Requirements

The headrests will feature automatic headrest adjustment, utilising a distance sensor on the car roof interior to know where the headrest should be located. The occupants can override this setup, with the driver's setting saved on their driver profile, to be restored whenever they return to the car.

The doors will feature a toggleable child lock, preventing the back doors from being opened from the inside, as well as a protection lock which automatically locks the doors from being opened from the outside after the car has started moving (this can also be toggled off if, for example, the driver is picking up a friend)



Sometimes, drivers will forget to dim their headlights for oncoming cars, resulting in dazzling them with the bright light. We aim to reduce the effect of our car doing this to other drivers, by implementing smart headlights that don't shine in the direction of other cars.[1] These headlights use the same technology that can be used to make rain appear mostly invisible.

While this solution doesn't immediately assist the driver of our car, over time, it is likely that other cars will pick up this technology, since the researchers are working on developing smaller versions (and anybody else on the road driving one of our cars would have this technology). It is difficult to develop a means of preventing dazzle for the driver of our car, since it is against the law to dim the windscreen to a level that would reduce this dazzle.

[1] http://spectrum.ieee.org/cars-that-think/transportation/systems/smart-headlights-make-driving-with-high-beams-safer

Comments

This task felt very different to previous tasks since there weren't many systems with UI to design and a lot of our ideas were mainly based around safety hardware. We enjoyed having a change in context for our systems and how the broadness of the topic allowed us to come up with wacky and cool ideas.

We were unable to come up with a single, all-encompassing final design because the safety features that we had were spread throughout the car. We tried to make each feature slide as specific as possible instead.

Some ill preparation for the last assignment when team members had other plans for the weekend and in the week did mean that the assignment bled into this week. We've tried to improve on this by working together to ensure that the work is done before the deadline date.

We did not allocate any tasks to individuals this week which may have stalled progress as many of us did not have the drive to write up information after the brainstorming. To improve our next assignment's work schedule, we could split our work evenly into tasks for each member of the group.

We feel our brainstorming for this task has improved, since we were more open to weird/crazy suggestions which we noted down and discussed - we have tried to be a little more sensible in previous weeks which may have restricted our design space

Mobile App

A-Team

Brainstorm

We considered both functional & non-functional requirements for the app in our brainstorm, we sectioned these in a Google Doc and all worked together discussing our ideas and adding them to the document.

We didn't rule out any ideas until the refined ideas stage as we wanted to keep our brainstorming moving quickly to get as many thoughts down as possible.

We thought of plenty of new features for the app that had not previously been considered, however, most of our ideas came from previous tasks where we stated that we wanted to be able to have control with an app.

To conclude our brainstorming phase, we decided to create a new document that we would fill with our final feature ideas. We thought through each feature from the initial brainstorm and discussed whether it would be well suited in our app and then copied it into our refined ideas document along with any extra details. As we moved features into the new document, we ensured that they were grouped by their use e.g. navigation, music/radio etc. With the ideas grouped, we found it easy to organise the app designs and imagine how each screen of the app would look.

Design Requirements

- Responsive layout
- *Easy to use™*
  - Accessible
    - Big buttons, font, etc.
    - High contrast colours
    - Things
    - Stuff
- *Aesthetically pleasing™*
  - Standard colour scheme
    - Connection to dashboard screens from earlier task
- Secure
- Efficient resource usage (CPU, me
- Efficient data structures, algorithms
  - Efficient mobile data usage
  - Binary? JSON? Gzip?
  - Well documented open-sou
- Contextual design

Features

- "Get car ready" feature
  - Choose the car, if you have
  - Being able to turn heating o
  - Air conditioning
  - Defrost windows
- User profiles
  - Settings
- Plan journeys (stops, route, music,
- Manage SatNav "favourites"
- Driving habits score history/tips vie
- Music features
  - Notified when playlists/mus

Navigation

Favourite Locations

- With nicknames (E.G. Johnas' school, Work, Wifes' lover, etc.)

Car tracking

- Useful for when the car is stolen
- Know where the car is parked
  - Perhaps a notification when detected 'walking back to your car' ren you where you parked it

Convoys

- Setup route
- Setup group
  - Share invite code through social media
  - Is a group of user profiles
    - Guest user used if no account available
- Member names taken from joining member's car's current account
- Member profiles taken from joining member's linked social media

Route Planning

- From home, transferred to car Over-the-air or using bluetooth/wifi when in vicinity if 1st method not available

Recent Searches

- Recent searches suggested when entering a destination

Music/Radio

Notifications for Finishing Tasks

- Synchronising podcasts, playlists, music

Playback Controls

- Play, Skip, Pause, Rewind, Fast-Forward

Requests

- Song to be played
- Song to be skipped
- Song to add to the queue

Research

For our internal research, we tried to find existing apps that accompany a car. We found a couple and discovered varying quality of features and usability from the apps. The VolksWagen Car-Net app is not very visually appealing but does have useful features, for example, it gave us the idea for scheduling some of our existing feature ideas.[1]
Jaguar has a system in their app in which many different apps can hook into the car's media system to provide custom functionality, for example, Spotify is able to play through the car directly from the Spotify app using 'Spotify Connect'.[2]

We also researched how the personas that we have designed might want to interact with the app and this allowed us to design features that we may have missed otherwise such as a permissions system for media controlling and sharing. We also realised that Agatha, one of our personas, would likely try to play the same songs on repeat so we want functionality in the app and car media system to allow the driver to be able to restrict this unwanted behaviour.

[1] http://www.volkswagen.co.uk/technology/car-net

[2] http://www.jaguar.co.uk/incontrol/connectivity/incontrol-apps.html

Design Lo-Fi

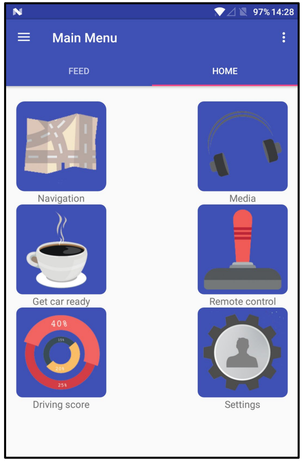
After having come up with all of the features that we want in the app, we tried to decide how we would like to arrange the screens to present these feature options to the user. We took a look at popular Android apps and default layouts and decided on layouts that we considered easy to use. Buttons on the main menu were designed to be large and have icons that gave clear differentiations.

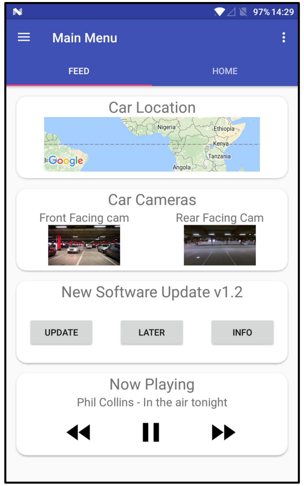
We decided to follow current styling standards used in industry to give the user a familiar experience, for example, we used flat design where we could, large icons and tried not to have any large swathes of text anywhere.

On the left, we have a scan of one of our lo-fi designs for the menu, it has been kept as simple as possible to increase ease of use.
The middle image shows the sidebar design, this was expected to be familiar to most users but with the customisation of having the profile details at the top of the bar.
The image on the right shows the lo-fi design for our feeds page, we decided that we would like it easily accessible from the home screen and so we have a tabbed layout with options at the bottom.

Design Hi-Fi

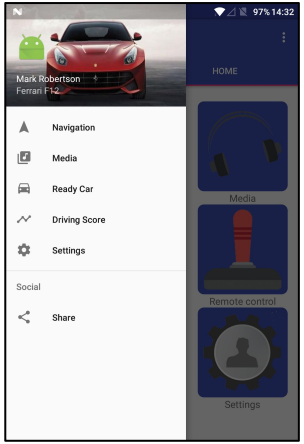
The home screen is one of the two primary tabs of the Main Menu that greet the user, it is a simple and intuitive view comprising only of buttons and labels for the main features of the app. It uses a vertically scrollable grid layout which makes it easily expandable if any additional features need to be included. The app also allows the user to reorder and hide the buttons to their preference to improve usability and satisfaction.





The feed is the other tab on the main screen, it consists of features which are displayed based on the context of where the mobile phone and the associated car is located. For example if the mobile phone is not in close proximity to the car it would include the feature of showing the location of the car. A variety of feature cards can be shown here, and can be interacted with in this screen to either directly control the feature - shown here with the music card - or take the user to the screen dedicated to that feature - if the user taps on the car location card it will take them to a screen to route them to their car, which is discussed on a later slide.

We designed a sidebar to allow users to be able to access all main sections of the app from any screen while in the app. We decided to show a picture of the car along with some simple account details at the top of the menu bar. This is to help the user know which car they are controlling. The android alien icon is a placeholder for a user profile picture in this example.
Icons for the menu items are simple and self-explanatory and there is a separate section for social media options.



Navigation

The Navigation submenu in the app is going to include:

- Add destination
- Schedule route
- Create new convoy
- Manage existing convoys
- Find my car

Add destination: Allows for the user to create a new route for near immediate execution. The interface will be made up of a search bar across the top of the screen and a map underneath. Once the user presses on the search bar the user's most common destination will be displayed until they start typing. Once they start typing, the app will try to match the input as a regular expression against the 100 most recently visited destinations. Once no match is found in the input string, a standard search will be performed when *return* or *enter* is pressed.

Schedule route: This feature will work similarly to *Add destination* with the main difference being that the user will be prompted for a time and date to schedule the new route for. This will then be transferred to the car and suggested to the user at the scheduled time.

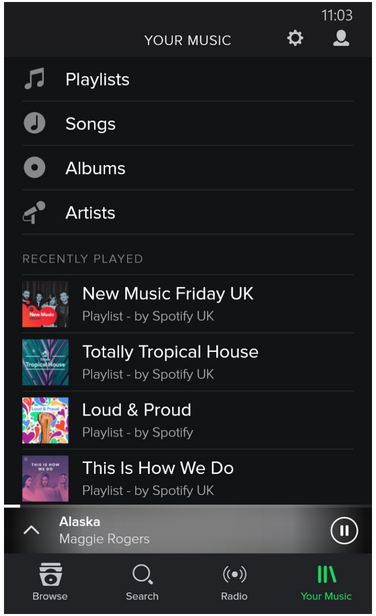
Create new convoy: Intuitively, this functionality allows the user to create a new convoy by first selecting a destination and later inviting other people to join. Once the convoy route is set up, a 6 character code will be displayed which can be used to allow others to join the convoy by typing the same code in their car's interface. The code will have a share button below it to easily invite people to the convoy via social media or text messaging.

Manage existing convoys: The user will be able to cancel/abandon a convoy, amend the route (given they are the creator or listed as administrator), add/request stops and invite participants.

Find my car: This feature will display a map with a pinpointed location of the car, with an option to calculate a route between the user's location and the car itself possibly on foot or by public transport. This will be useful for situations in which the driver has forgotten where they parked.

Music/Radio

The app will also feature the ability to control the sound being played on the car's stereo.
On entering the music section of the app you are greeted with a screen showing what is currently playing. Users interact with the system differently depending on whether they are listening to the radio or music which is stored locally or streamed from services such as Spotify, Apple Music, Amazon Music, Google Play or others.



In the case of music being played, users will have the usual playback controls and options to add and skip songs in the current queue. Actions such as adding items to the queue and skipping tracks depend on the privilege of the user, based on their user profiles (e.g. Drivers have high privilege, children have low privilege). This is used to stop users changing music abruptly. High privileged users will be able to perform actions with no problems, however low privileged users will have to request an action for a high privileged user to approve it for it to happen.

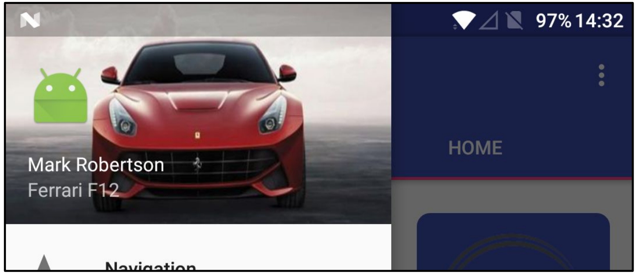
Users will be able to select radio stations, search and browse programmes with the ability to favourite programmes so the user is notified when they go live.

There will also be phone notifications for when the car has completed synchronising podcasts, playlists and music with other storage media such as their mobile phones and recognised home networks which the car will connect to when parked up and in range outside (mentioned in previous weeks).

User Profiles

For a lot of our designs, we have utilised the fact that the driver, and often passengers, have their preferences saved on their personal user profile. These profiles are managed by the phone app; with the phone app being linked to their profile and offered as a means of identification while the phone is linked to the car.

When the app is first launched, the user is prompted to log in or sign up with a username, email address, password, address and phone number. They also have the choice to log in using a social media account, or can connect to one at a later time - with the option to sync their user profile picture with their social media profile picture.



This user profile provides a link to some of the details unique to their user, found on their phone, such as synchronising with their contacts and calendars, and retrieving and storing an emergency profile for the user - containing name, address, blood type, allergies, medications, organ donor status, medical notes as emergency information pertaining to the user, as well as emergency contact details of designated contacts.

These profiles store settings and preferences for the user, including preferred temperature, positions of seat and steering wheel, music preferences and SatNav preferences. These settings can also be linked to a certain car if the user has access to multiple cars.

User profiles will also have privileges associated with them where users with high privilege will have the ability to use certain features such as locking and unlocking doors, starting the car and control of music.

Control Car Features

Our app will be able to control certain hardware features in the car such as setting the car’s temperature, toggling the headlights or heated seats and defrosting windows. Many of these features will be packaged into a section of the app available from the main menu, which will allow users to get the car ready for their journey using saved settings. For example, in winter, a driver might want to get their car ready for a journey by defrosting the windows and heating up the car and seats. Since some of these features will run the battery flat, the car will be able to automatically turn everything off if the car is not started within an hour or the battery begins to run low.

System Updates

The car entertainment and navigation system will be updated through delta updates, this is so that updates can be downloaded quicker than full system images and use less bandwidth.

It will be possible to download updates through the car app and transfer them to the car to be installed. The app will notify the user that an update is available through phone notifications and the app’s feed view.

If the user clicks either of the notifications, they will be taken to a screen showing the car’s current software version, the update’s version and the update changelog. There will be buttons for the user to either ‘Dismiss’ the update or ‘Download/Install’

Driving Score

In a previous task, we outlined that one way to increase safety for drivers would be to encourage them to improve their driving ability by scoring their driving habits and offering suggestions based on their strengths and downfalls.

Our app would integrate with data from the car to be able to show a history of past driving scores with some attractive graphs and informative graphs as well as the personalised driving tips specific to the user and car.

The car can optionally assist during driving, for those who enable the option in their account, with small audible notices such as beeps or spoken word to advise the driver on improvements to their driving- these can also be kept track of in the app.

Specific stats would be available to help the driver to track their progress over time, using previous data that has been cached in their account on the cloud servers. Statistics such as how often the driver has been too close to the car ahead on a journey can be demonstrated, as an example.

Social Media
The user of our app will be able to use their social media accounts to improve their experience of the app by being able to view their friends' driving scores provided that they have chosen to share them as well as being able to share data used in the app for telemetric analysis to improve the driving suggestions.

Examples of shareable data from the app include journey statistics, routes and driving behaviours on those particular routes with the possibility to expand with additional features in the future via software updates.

Social media accounts can also be used to link convoy members to their profile pictures and similar user-friendly information.

Car Security

During the design of our app, we decided that it would be very useful for the car's security features to be able to connect with the accompanying app to improve both ease of use and the safety of the user's possessions.

The first of our car security features are relating to theft prevention and recovery of the car. If the car alarm sounds, the car will notify our cloud server of this event which will then be pinged down to the user's phone, meaning that they are then aware that their car alarm is going off and that they can react accordingly. Since we have previously discussed that our car will have video cameras around its body, we think that it would be useful for the users of our app with the appropriate permissions to be able to remotely access video feeds from this camera, this could be useful to check whether anything bad has happened to cause the car alarm to go off for example. The app would be able to silence, enable and disable the car's alarm, useful for a false alarm or for if people are being left in the car. The final active car protection/recovery feature is that the car can be tracked via the app, this would, of course, be very useful for if the car was stolen. The tracking feature will share the same screen in the security section as is used for the "Find my car" feature, used for tracking where a driver parked.

Our car is designed to have a more advanced locking system, allowing for greater flexibility. The mobile app will be able to tell the user whether the car is locked or unlocked and notify them if they walk away from the car without locking it. In addition to this, the user will, in fact, be able to unlock the car doors using the phone's NFC capabilities, also requiring a passcode or fingerprint scan on the phone. Hobbyists are already creating systems to unlock cars via NFC, ours could be the first sleek, integrated NFC car unlocking system.[1]

Since many user profiles might exist for a car, the car unlocking feature is also an improvement on typical physical keys since access to the car can be granted on a per-user basis. The permissions system for users would identify users by their email or username and allow them different levels of access to the car. For example, a user could be granted permissions to fully control the music system or only to request song changes/additions, to lock and unlock the car or to be able to start the car.

[1] http://bit.ly/2n3Jmzr

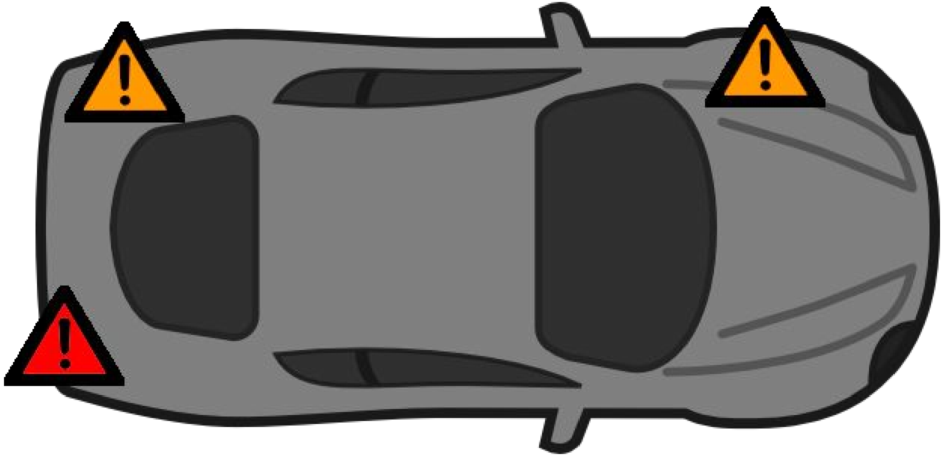
Car Stats

A section of the app will detail some statistics about the car, some of which will be specific to the logged-in user.

The status and wellbeing of the car will be shown here, displaying system errors and warnings, offering descriptions and resolution tips for these as well as any given error code that the user can input. These warnings will show until they are fixed, and any new errors that occur will trigger a notification to the phone. As well as errors, the levels of various storages can be tracked such as fuel, water, oil and battery levels, which trigger a notification when they are low. Service history and related recurring events are shown here too, such as MOT date, and expiry dates of insurance and tax. The user enters these dates manually and receives a notification when the time is coming close.

The system faults will be displayed as highlighted sections of a top-down view of the car, where these sections can be tapped to show the extra information about the fault. This design is demonstrated in the given picture, showing warnings for a rear light failure, low fuel level and a check brakes warning.

Travel statistics are also shown in this section, with information on mileage and average MPG ratings. These can be toggled to show values for the car's lifetime, or specific to just the logged-in user.



Car Remote Control

Our app will also feature the ability to control the car remotely. It is intended primarily to be used for parking your car, or maneuvering a tight turn when it is difficult for the driver to see where the car is going from the driver seat.
Drivers will be able to exit their vehicle, launch the remote control system and guide their car from a different point of view to manoeuvre their car with ease.

During remote control, some crucial safety precautions are put in place: A car can only be controlled to a maximum speed of 5 miles per hour and the device must be within a 10-metre radius from the vehicle otherwise it will stop automatically to prevent accidental crashes. The onboard computer with constantly monitor and react to the car surroundings, overriding user input, allowing a safe and collision free operation

The interface for controlling the car will consist of a large steering wheel in the centre of the screen to steer the car. On the left, there is toggle for controlling the driving direction, and on the right there is a linear slider to vary the speed of the car.

In our research, we found that Jaguar Land Rover have developed a working prototype of remotely controlling a vehicle via a simple *game-like* interface in a smartphone app

http://www.landrover.com/experiences/news/jlr-remote-control-range-rover-sport.html

Communication & Pairing

Wherever possible, communication between the phone and car is via the WiFi network broadcast from the car as it provides the fastest and most stable connection whilst keeping battery usage to a minimum. Conveniently, the WiFi network doubles as an internet access point for the phone as well as a communication channel for the app’s API. It is also recommended that devices are paired via Bluetooth too as a redundant channel (or for primary communication for older devices) which can also be used for signalling and small amounts of data transfer.

Initial Pairing
During initial setup, 2 pairing modes are available to the user:

- WiFi & Bluetooth via NFC (place phone on charging pad)
- Manual pairing using PIN number within the app

Paired devices that are trusted by the car can be allowed on a per-device basis in the car settings to unlock the car from a close proximity and remotely administer and control the car if desired.

NFC (RFID)
As well as during the pairing process, the NFC protocol doubles up as a secure and close-range protocol that can be used to unlock the car with any paired and trusted device like a smartphone or smartwatch using the readers in the doors.

Cellular connections (3G, 4G, 5G etc)
All cars should be fitted with in-built cellular radios to provide an always-on internet connection to both the car and any devices paired with it. The smartphone app also provides the functionality to share it’s own cellular connection with the car for areas of poor coverage for backup and tracking purposes only; it cannot be used as an upstream for other devices connected to the car to reduce data usage and ensure security.

WiFi Network Sharing
The app has functionality to transfer recognised WiFi network credentials from the phone onto the car computer to allow the car to connect and synchronise with cloud servers when they are in range. *(Assumes appropriate mobile OS support)*

Technical details

All data will be stored or at least cached within the car, and a smaller subset also cached within the app primarily for data availability as it cannot always be assumed there is a network connection. All data will be permanently stored on cloud servers as a long-term & offsite persistent backup. The mobile app will use SQLite for primary data storage and fallback to a hierarchical file structure where required.

API calls between the car, the app and cloud servers[1] will use HTTP as it has a universal compatibility with a JSON schema for exchanging data- all traffic is served over a TLS connection.

Encryption

It is required that all car systems authenticate themselves with a client X509 certificate signed by the respective car manufacturer and any other authoritative parties. The car system must always use this client certificate when negotiating TLS connections to allow the recipient to verify the car's identity and authenticity.

The smartphone app does not require a client certificate as the only purpose it serves is as a 'dumb' read/write endpoint and will never store authoritative data about the car or user account.

It goes without saying that any connection which is negotiated with an invalid certificate should be immediately terminated as it's likely the connection is compromised or the car isn't genuine.

Storage

All data served from the app is stored on a cloud server or is served directly from the car (for data like live video, audio and information streams) and is only stored on the phone as a cache to improve usability, availability and loading times. Map routes are cached within the app after being generated and are deleted immediately when any of the following occur:

- It has been followed
- It is flagged as stale from another device
- A change is made elsewhere (the newer version replaces it)
- Three days pass (since initial route computation)

[1] https://d21ii91i3y6o6h.cloudfront.net/gallery_images/from_proof/3442/large/1418280711/die-cut-stickers.png

Comments

We enjoyed this task more than some of the others since there was a fair amount of software design to be done.
Since we're all taking the 'Ubiquitous Computing' module, we all have some knowledge of how to design, build and run Android applications. We decided that after initial sketches of how we thought our app might look, the easiest way to make high fidelity designs would be to actually develop the activity layouts for an Android app. We collaborated on the example app, keeping our code in a GitHub repository.[1]

The app mocks basic navigational functionality and appearance of how we envisioned it would look and it helped us understand better how it would interwork with the car system as a whole. It prompted much discussion amongst the team concerning how it would look and function as well as making us consider good quality UI design and competent app usability.

One of our team members was ill for a large part of the week after our brainstorming phase was complete. We pushed on and were able to complete a lot of work regardless since our ideas seemed very clear to us.

[1] https://github.com/ChrisLane/car-control